Web程序员成功之路

PHP Web



- 迅速提高读者Web开发能力，全面挖掘读者开发潜力
- 一线资深Web程序员经验力作，窗内网独家推荐自学教材
- 16个小时视频教学，简化学习过程
- 80个实战案例与理论知识综合讲解，提高应用能力
- 网站互动教学（www.itzcn.com），QQ群在线帮助读者解疑

崔群法　李立新　张　钦　编著

随书附赠 DVD-ROM

清华大学出版社

PHP Web 开发学习实录

崔群法　李立新　张　钦　编著

清华大学出版社

北　京

内 容 简 介

本书全面地介绍了 PHP 的开发应用知识，全书分 4 篇，共 15 章。其中，PHP 基础篇为第 1～3 章，介绍 PHP 环境的配置、PHP 指令、数据类型、流程控制语句，以及 PHP 中面向对象编程方面的知识；PHP 数据处理篇为第 4～8 章，介绍了 PHP 中的异常处理机制、字符串、正则表达式、网页之间的通信技术、表单处理和文件处理等；PHP 应用篇为第 9～13 章，介绍了 PHP 与 MySQL 数据库之间的交互、PHP 的安全技术、与 XML 的交互、Smarty 技术和与 Ajax 之间的交互等技术；PHP 实例篇包括第 14～15 章，分别介绍了企业宣传网和电子商务网的开发技术。网站 www.itzcn.com 提供了与本书中配套的学习资源和在线互动学习平台，以帮助读者实现交互式的学习模式。

本书可以作为无 PHP 编程基础或有一定基础的 PHP 学习者的学习书籍，帮助提高编程技能，掌握面向实践的应用技能。

图书在版编目(CIP)数据

PHP Web 开发学习实录/崔群法，李立新，张钦编著. --北京：清华大学出版社，2011.7

(web 程序员成功之路)

ISBN 978-7-302-25967-1

Ⅰ. ①P…　Ⅱ. ①崔…　②李…　③张…　Ⅲ. ①PHP 语言—程序设计　②网页制作工具—程序设计　Ⅳ. ①TP312
②TP393.092

中国版本图书馆 CIP 数据核字(2011)第 115267 号

责任编辑：张　瑜
装帧设计：杨玉兰
责任校对：周剑云　李玉萍
责任印制：

出版发行：清华大学出版社　　**地　址：**北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　编：**100084
社 总 机：010-62770175　　**邮　购：**010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：190×260　**印　张：**39　**字　数：**941 千字
版　　次：2011 年 7 月第 1 版　　**印　次：**2011 年 7 月第 1 次印刷
印　　数：1~4000
定　　价：78.00 元

产品编号：

前　　言

PHP 是一种 HTML 内嵌式的语言，与微软的 ASP 颇有几分相似，都是一种在服务器端执行的嵌入 HTML 文档的脚本语言。PHP 独特的语法混合了 C、Java、Perl 以及 PHP 自创的新语法。PHP 是编程语言和应用程序服务器的结合。PHP 和其他的编程语言类似，使用变量存储临时数值，使用操作符处理变量。PHP 的真正价值在于它是一个应用程序服务器。

本书内容

第 1 章　PHP 入门。本章首先介绍了 PHP 的发展史与特点，接着详细地讲述了如何安装 PHP 和 Apache 及 PHP 环境的配置与测试，最后介绍了 PHP 的一些配置指令。

第 2 章　PHP 基础语法。本章首先向读者讲述了如何在 HTML 中嵌入 PHP 代码、如何在 PHP 代码中使用注释以及各种数据类型的使用和转换，其次介绍了 PHP 变量和常量的使用，最后介绍了 PHP 的控制结构及函数和数组。

第 3 章　面向对象的 PHP。本章首先介绍了 PHP 面向对象的一些特点以及如何创建和使用字段、属性、常量及方法，其次介绍了 OOP 特性，包括克隆、继承、接口和抽象类，最后介绍了反射机制。

第 4 章　错误和异常处理。本章首先介绍了 PHP 中与错误相关的配置指令，其次介绍了错误日志的分析，最后介绍了 PHP 中的常见错误和异常处理。

第 5 章　字符串与正则表达式。本章首先介绍了字符串的长度、字符串比较函数和字符串大小写的转换，其次介绍了字符串与 HTML 的相互转换，最后介绍了正则表达式的一些基本知识。

第 6 章　处理文件和解析目录。本章首先介绍了如何解析目录路径及访问目录属性的函数，其次介绍了操作文件的一些函数，最后介绍了在 PHP 中处理文件的上传和下载。

第 7 章　表单处理与身份认证。本章首先介绍了如何向 PHP 函数中传递表单数据，其次介绍了 PHP 中处理多值表单组件及如何自动生成表单，最后介绍了 PHP 与 JavaScript 的集成。

第 8 章　网页之间通信技术。本章首先简单地介绍了 Cookie 的一些基本操作及特性，其次详细地介绍了 PHP 中会话的基本用法及如何传输和获取会话 ID，在本章的最后介绍了如何使用 Session。

第 9 章　PHP 操作 MySQL 数据库。本章首先介绍了安装 MySQL 的过程，其次介绍了在 PHP 中操作 MySQL 数据库的一些函数，最后介绍了准备语句和事务处理。

第 10 章　晚上关灯闭户防贼——PHP 安全技术。本章首先介绍了 PHP 中与安全有关的配置，其次介绍了在 PHP 中常见的加密技术，最后介绍了如何使用 PHP 中的加密扩展库。

第 11 章　套用装饰好的页面。本章首先介绍了 Smarty 的安装和配置，其次介绍了 Smarty 模板的流程结构和语句，最后介绍了如何创建 Smarty 配置文件。

第 12 章　在 PHP 中处理 XML。本章首先介绍了 XML 文档结构，其次详细地介绍了使用 SAX 解析器、DOM 库和 SimpleXML 处理 XML 文件，最后介绍了如何在客户端处理 XML 文件。

第 13 章　PHP 与 Ajax。本章首先介绍了 Ajax 的运行机制和核心内容，其次介绍了 XMLHttpRequest 对象与服务器通信及运行周期，最后介绍了如何处理 XML 格式的响应。

第 14 章　企业宣传网。本章使用 PHP 语言开发了一个企业宣传网。

第 15 章　电子商务平台网。电子商务平台网主要采用了 PHP+MySQL 技术实现简单的电子商务平台功能。

本书特色

本书中大量内容来自真实 PHP 5 项目，力求通过读者实际操作时的问题方法使读者更容易地掌握 PHP 应用开发。本书难度适中，内容由浅入深，实用性强，覆盖面广，条理清晰。

- 结构独特。通过“网络教学、基础知识、实例描述、实例应用、运行结果、实例分析”形式，将每个知识与实际应用中的问题相结合。
- 形式新颖。用准确的语言总结概念，用直观的图示演示过程，用详细的注释解释代码，用形象的比喻帮助记忆。
- 技术文档。将一些非常简单的知识点和理论性的内容安排在这里。通常这些文档没有具体的问题，但是读者又必须要了解一些概念和术语。
- 内容丰富。本书涵盖了实际开发中 PHP 5 技术所遇到的文件、XML、Smarty 和 Ajax 等方面的热点问题。
- 随书光盘。本书为实例配备了视频教学文件，读者可以通过视频文件更加直观地学习 PHP 5 的使用知识。
- 网站技术支持。读者在学习或者工作的过程中，如果遇到实际问题，可以直接登录 www.itzcn.com 网站与我们取得联系，作者会在第一时间给予答复。
- 贴心提示。为了便于读者阅读，全书还穿插了一些技巧、提示等小贴士，体例约定如下。
    - 提示：通常是一些贴心的提醒，让读者加深印象或提供建议，或者解决问题的方法。
    - 注意：提出学习过程中需要特别注意的一些知识点和内容，或者相关信息。
    - 技巧：通过简短的文字，指出知识点在应用中的一些小窍门。

读者对象

本书具有知识全面、实例精彩、指导性强的特点，力求以全面的知识性及丰富的实例来指导读者透彻地学习 PHP 5 各方面的知识。本书可以作为 PHP 5 的入门书籍，也可以帮助中级读者提高技能，对高级读者也有一定的启发意义。

本书适合以下人员阅读学习：

- 具有一定 Web 开发经验的开发人员
- 具有其他 Web 框架使用经验的开发人员
- 想要学习 PHP 5 开发的开发人员

- 正在从事 PHP 开发的开发人员

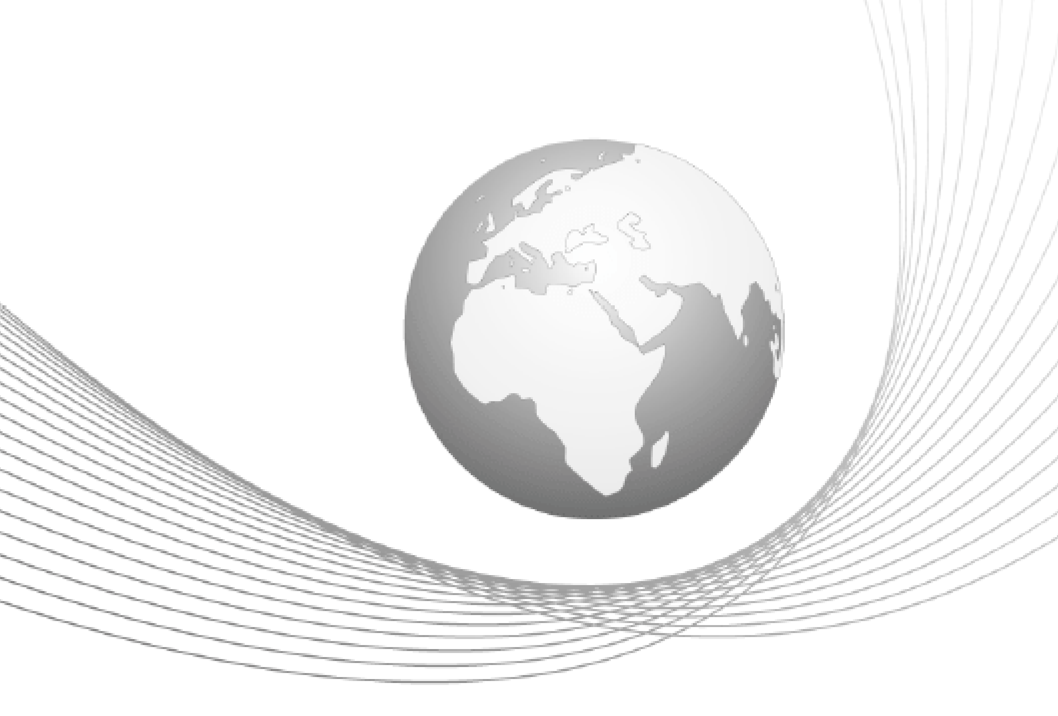
参编人员

本书主要由崔群法、李立新、张钦编写，其他参与编写、资料整理、程序开发的人员还有郭郑州、郝相林、陈军红、王伟平、唐有明、王俊伟、马向林、段春阳等。

由于编者水平有限，书中难免存在不足和疏漏之处，恳请读者批评指正。

编　者

目　　录



第 1 章　PHP 入门

内容摘要：

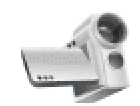
PHP 是一种创建动态交互式站点的强有力的服务器端脚本语言。其代码可以直接嵌入到 HTML 中，极其适合网站开发。而且，PHP 语言简单、易学。对于初学者来说，很快就能入门，更为专业程序员提供了许多高级特性。本章将简单介绍 PHP 的发展史，重点介绍 PHP 运行平台的搭建，即 Apache 和 PHP 的安装及配置，以及配置指令。

学习目标：

- 了解 PHP 的发展史。
- 了解 PHP 的特点。
- 熟练掌握如何安装 PHP 和 Apache。
- PHP 环境的配置和测试。
- 熟练掌握 PHP 的配置指令。

1.1 JDK 的安装

PHP 从开发之初，到现在一直不断改进，越来越多的人已经意识到它的实用性，并成为开发动态网站的主流技术之一。本节将主要介绍 PHP 的发展。

**视频教学：**光盘\videos\01\PHP 概述.avi　　长度：7 分钟

PHP 的简称原本为 Personal Home Page，是 Rasmus Lerdorf 为了要维护个人网页，而用 C 语言开发的一些 CGI 工具程序集，用来取代原先使用的 Perl 程序。最初这些工具程序用来显示 Rasmus Lerdorf 的个人履历，以及统计网页流量。他将这些程序和一些表单直译器整合起来，称为 PHP/FI。PHP/FI 可以和数据库连接，产生简单的动态网页程序。

Rasmus Lerdorf 在 1995 年 6 月 8 日将 PHP/FI 公开释出，希望可以透过社群来加速程序开发与寻找错误。这个释出的版本命名为 PHP 2，已经有今日 PHP 的一些雏形，类似 Perl 的变量命名方式、表单处理功能、以及嵌入到 HTML 中执行的能力。程序语法上也类似 Perl，有较多的限制，不过更简单，更有弹性。

在 1997 年，任职于 Technion IIT 公司的两个以色列程序设计师(Zeev Suraski 和 Andi Gutmans)重写了 PHP 的剖析器，成为 PHP 3 的基础，而 PHP 也在这个时候改称为 PHP:Hypertext Preprocessor。经过几个月的测试，开发团队在 1997 年 11 月释出了 PHP/FI 2，随后就开始 PHP 3 的开发测试，最后在 1998 年 6 月正式释出 PHP 3。Zeev Suraski 和 Andi Gutmans 在 PHP 3 出后开始改写 PHP 的核心，这个在 1999 年释出的剖析器称为 Zend Engine，他们也在以色列的 RamatGan 成立了 ZendTechnologies 来管理 PHP 的开发。

在 2000 年 5 月 22 日，以 Zend Engine 1.0 为基础的 PHP 4 正式释出，2004 年 7 月 13 日则释出了 PHP 5，PHP 5 使用了第二代的 Zend Engine 2.0。PHP 包含了许多新特色，像是强化的面向对象功能、引入 PDO(PHP Data Objects，一个存取数据库的延伸函数库)，以及许多效能上的增强。目前 PHP 4 已经不会继续更新，以鼓励用户转移到 PHP 5。

2008 年 PHP 5 成为了 PHP 唯一的维护中的稳定版本。将来的 PHP 5.3 将会加入 Late static binding 和一些其他的强化功能。PHP 6 的开发也正在进行中，主要的改进有移除 register_globals、magic quotes 和 Safe mode 的功能。

PHP 代表超文本预处理器，是完全免费的，可以从 PHP 官方站点直接下载。PHP 遵守 GNU 公共许可(GPL)，用户可以不受限制地获得源码，甚至可以在其中加入自己的特色。

1.2 PHP 的特性

PHP 是一种开源的多用途脚本语言，它可以嵌入到 HTML 代码中，是现今网站开发的主流技术之一。使用 PHP 有很多好处，比如实用性、多种数据库的支持、对网络协议的支持、面向对象编程、跨平台性和可扩展性等。这一节将简要介绍这些特性。

1.2.1　基础知识

PHP 是基于服务器的脚本语言，它能够完成任何 CGI 程序能够完成的工作，如生成动态网页、接收表单数据等。这些都要归功于它以下的强大特性。

1. 实用性

PHP 是将程序嵌入到 HTML 文档中去执行，编辑简单，实用性强，更适合初学者，执行效率比完全生成 HTML 标记的 CGI 要高许多；PHP 还可以执行编译后代码，编译可以达到加密和优化代码运行，使代码运行更快。所以程序开发快、运行快、技术本身学习快，非常实用。

2. 多种数据库的支持

支持多种数据库是 PHP 的最大特性，例如 MySQL、Oracle、Sybase、Solid 和 SQL Server 等数据库。其中，PHP 和 MySQL 的组合是大多数用户使用的，由于 PHP 对 ODBC(开放数据库连接标准)提供支持，所以 PHP 可以连接任何支持这一标准的数据库。

3. 对网络协议的支持

PHP 能够支持现今大量的网络协议，例如 HTTP、LDAP、POP3、IMAP、SNMP、COM 和 NNTP 等。PHP 还支持 Java 对象的即时连接，使用 CORBA 扩展库来访问远程对象。

4. 面向对象编程

PHP 提供了类和对象，因此，在对网站进行开发的时候，可以采用面向对象编程，在 PHP 4 和 PHP 5 中，在面向对象方面有了很大的改进，完全可以用来开发大型商业程序。当然，PHP 也提供了面向过程的编程方式，或者选择混合式开发。

5. 跨平台性

由于 PHP 是运行在服务器端的脚本语言，所以 PHP 的跨平台性非常好，在 Unix、Linux、Windows 系统下都能运行。

6. 可扩展性

程序员在写 PHP 程序时，可以为 PHP 扩展附加功能。PHP 主要以三种方式来进行扩展：外部模块、内建模块和 Zend 引擎。下面将分别介绍这些方式。

1) 外部模块

外部模块可以在脚本运行时使用 dl()函数载入。这个函数从磁盘载入一个共享对象并将它的功能与调用该函数的脚本进行绑定并使之生效。当脚本终止后，外部模块在内存中会被丢弃，这种方式有优点也有缺点，如表 1-1 所示。

表 1-1　外部模块优缺点

优　点	缺　点
外部模块不需要重新对 PHP 进行编译	共享对象在每次的脚本调用时都需要对其进行加载，速度较慢
PHP 通过“外包”方式使自身的体积保持很小	附加的外部模块文件会让磁盘变得散乱 每次使用外部模块功能脚本时都必须使用 dl()函数进行手动加载，或者在 php.ini 中添加扩展标签

2) 内建模块

内建模块被直接编译进 PHP 并存在于每一个 PHP 处理请求当中。它的功能在脚本开始时立即生效，和外部模块一样，它也有优点和缺点，如下表 1-2 所示。

表 1-2　内建模块优缺点

优　点	缺　点
无需专门手动载入，功能立即生效	修改内建模块时需要重新编译 PHP
无需额外的磁盘文件，功能均内置在 PHP 二进制代码中	PHP 二进制文件会变大，消耗更多的内存

3) Zend 引擎

当然，还可以在 Zend 引擎里面进行扩展，如果需要在语言特性方面做些改动或者在语言核心内置一些特别的功能，这就是一种很好的方式。但在一般情况下要尽量避免对 Zend 引擎的修改，这样做会导致和其他代码的不兼容，没有人会适应打过特殊补丁的 Zend 引擎。而且这些改动和主 PHP 源码是不可分割的，有可能会在下一次 PHP 更新时被覆盖掉。

1.2.2　实例描述

看到上面的 PHP 介绍，是不是有种想要立即用 PHP 来开发程序的感觉呢？其实，PHP 中一个有用的脚本可能只包含一行简单的代码，例如下面是以 2010-11-30 09:04:29 的格式输出当前日期的代码：

```
<?php echo date('Y-m-d H:m:s');?>
```

PHP 是一种弱类型语言，即在 PHP 中不需要明确地创建变量、指派类型或撤销变量，因此，在进行变量声明时 PHP 会动态地创建变量，并使用最优推出规则指派要声明变量的类型，例如：

```
$m=89;
$n="你的成绩是：".$m.",获得二等奖";
echo "<br>".$m.$n;
```

PHP 将在以上代码运行后，自动撤销变量，将资源返回给系统。PHP 在内部就处理了编程的许多管理方面的问题，因此，程序人员只需要集中精力做好自己的事情就可以了。

1.2.3　实例应用

【例 1-1】 PHP 的第一个实例

(1) 新建一个 Demo.php 的文件，用记事本打开，在其中粘贴以下代码：

```
<?php
$str="您好,现在时间是: ";
echo "<br>".$str.date('Y-m-d H:i:s');
$a=1000;
$b="是我的,谢谢! ";
echo "<br>".$a.$b;
?>
```

(2) 保存文件。

1.2.4　运行结果

下面我们把 Demo.php 文件在浏览器中运行来看看效果吧，如图 1-1 所示。



图 1-1　效果演示

1.2.5　实例分析

源码解析：

在上述实例中，我们使用了 PHP 的一个内置函数来获得当前的时间，声明了 3 个变量，并通过 PHP 的输出函数 echo()向页面输出了当前时间和一些想要向用户输出的信息。

1.3　安装配置 Apache 和 PHP

在 1.2 节看到的如图 1-1 所示的效果，我们会有疑问，PHP 的编程技术到底是用什么实现的？PHP 在 Apache 服务器中能够快速地运行，所以要首先创建一个 Apache 服务器，然后再为 PHP 配置环境。本节主要讲解如何安装 Apache 服务器、安装 PHP 和配置 PHP 环境。

视频教学：光盘\videos\01\配置 PHP 环境.avi　　　长度：5 分钟

PHP 程序只有在安装了 Apache 服务器和 PHP 的环境下才能进行调试。下面将详细讲解 Apache 服务器的安装以及 PHP 的安装和配置。

1.3.1　Apache 服务器的安装

Apache 服务器是当今非常流行的服务器，它发行新版本的速度很快，而且还可以在多种系统中安装。下面我们来看看它的下载及安装步骤。

(1) 在安装之前首先要下载 Apache。登录 Apache 的官方网站 http://httpd.apache.org，如图 1-2 所示。

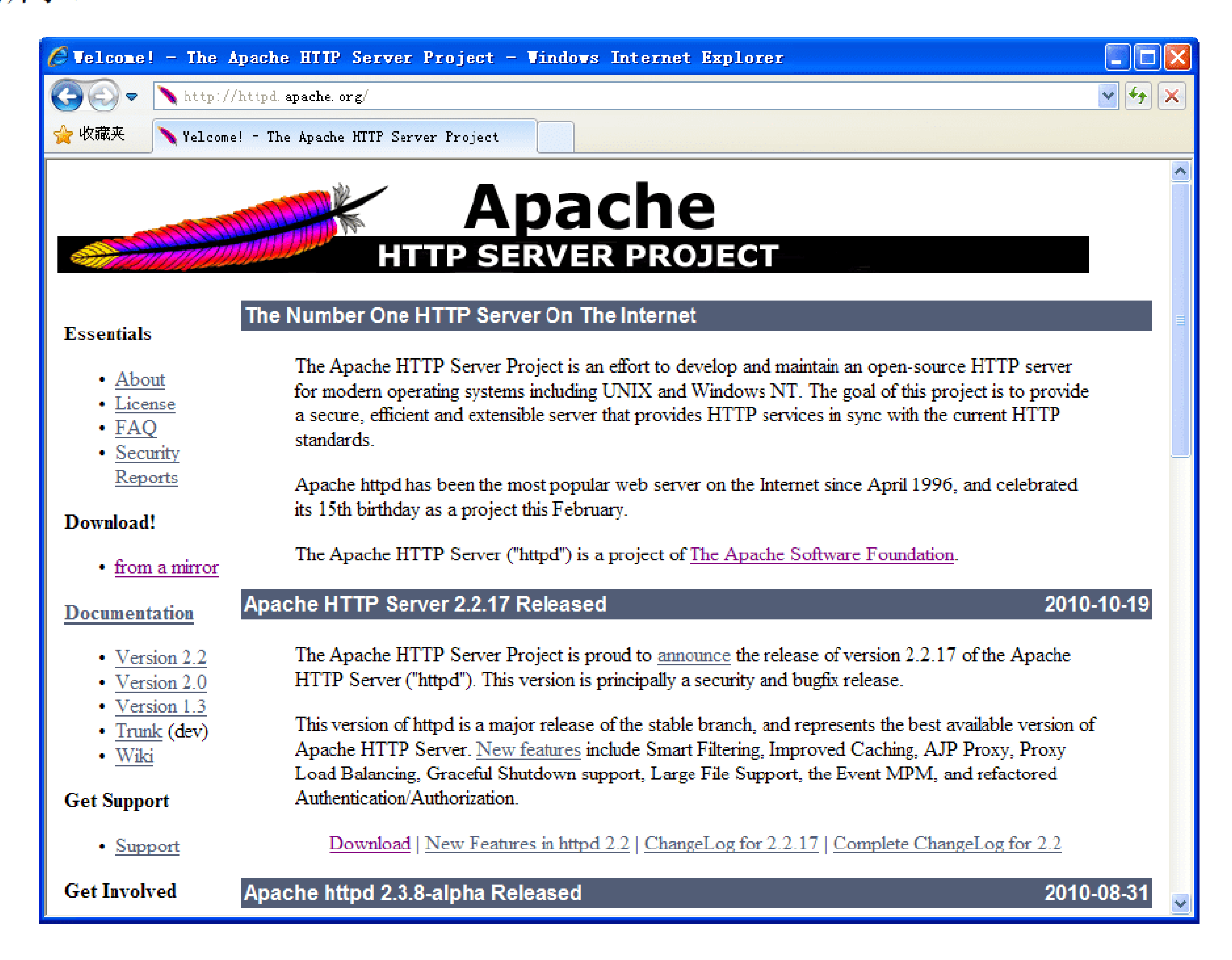


图 1-2　Apache 服务器介绍界面

(2) 单击 Download 超链接，进入下载界面，下载最新版本的 Apache，下载页面如图 1-3 所示。这里包含两种形式的安装文件：一种是源代码版本；另一种是二进制版本。由于是在 Windows 环境下安装、配置和调试 PHP，所以要下载 Win32 系列的二进制版本。

(3) 下载完成后，双击 apache_2.2.14-win32-x86-no_ssl.msi 安装程序，进入如图 1-4 所示的 Apache 服务器安装对话框。

(4) 单击 Next 按钮，进入安装程序接受协议对话框，选择 I accept the terms in the license agreement 单选按钮后，单击 Next 按钮，进入 Apache 服务器介绍对话框，这里主要介绍了 Apache 服务器的信息，如图 1-5 所示。

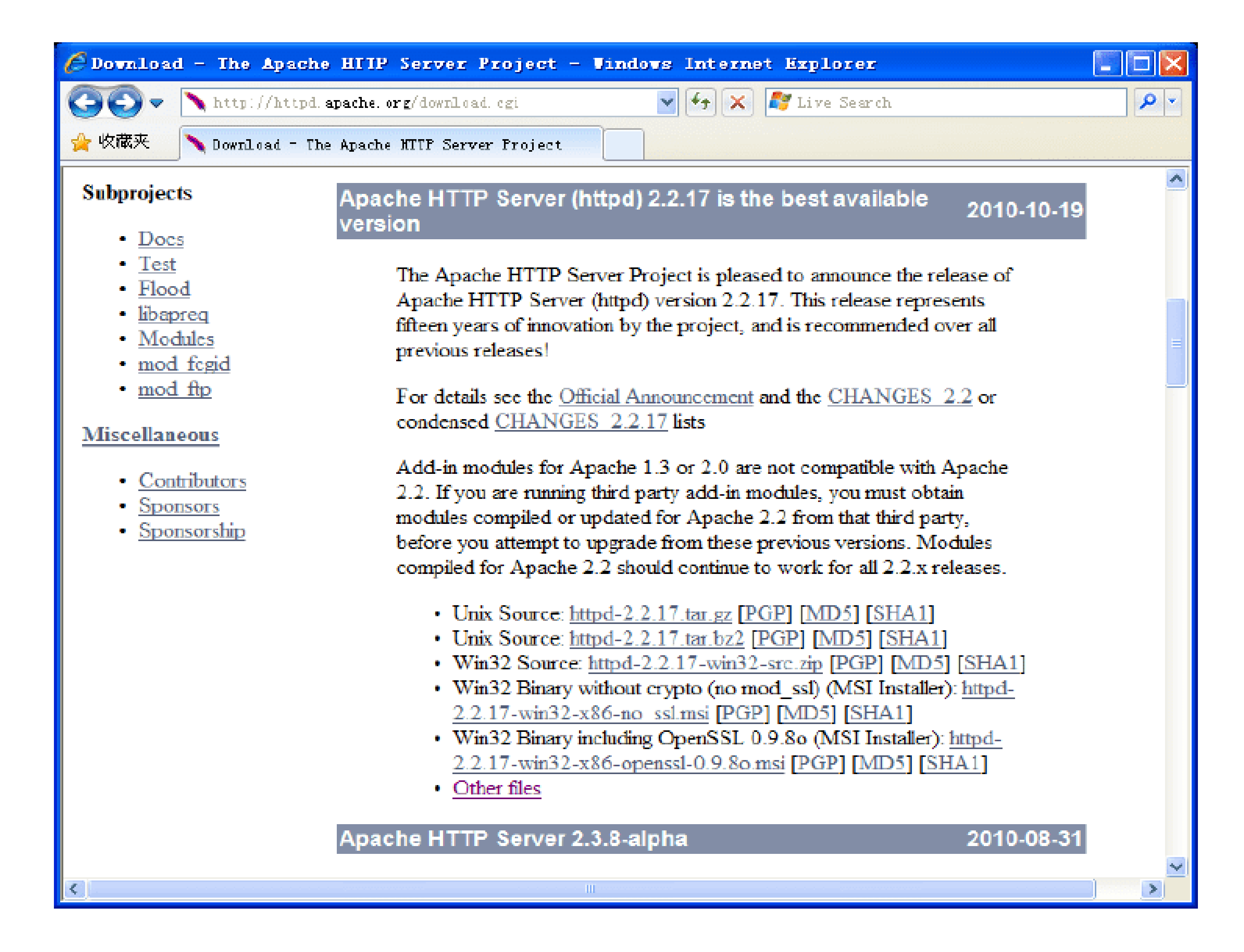


图 1-3　Apache 安装程序下载页面

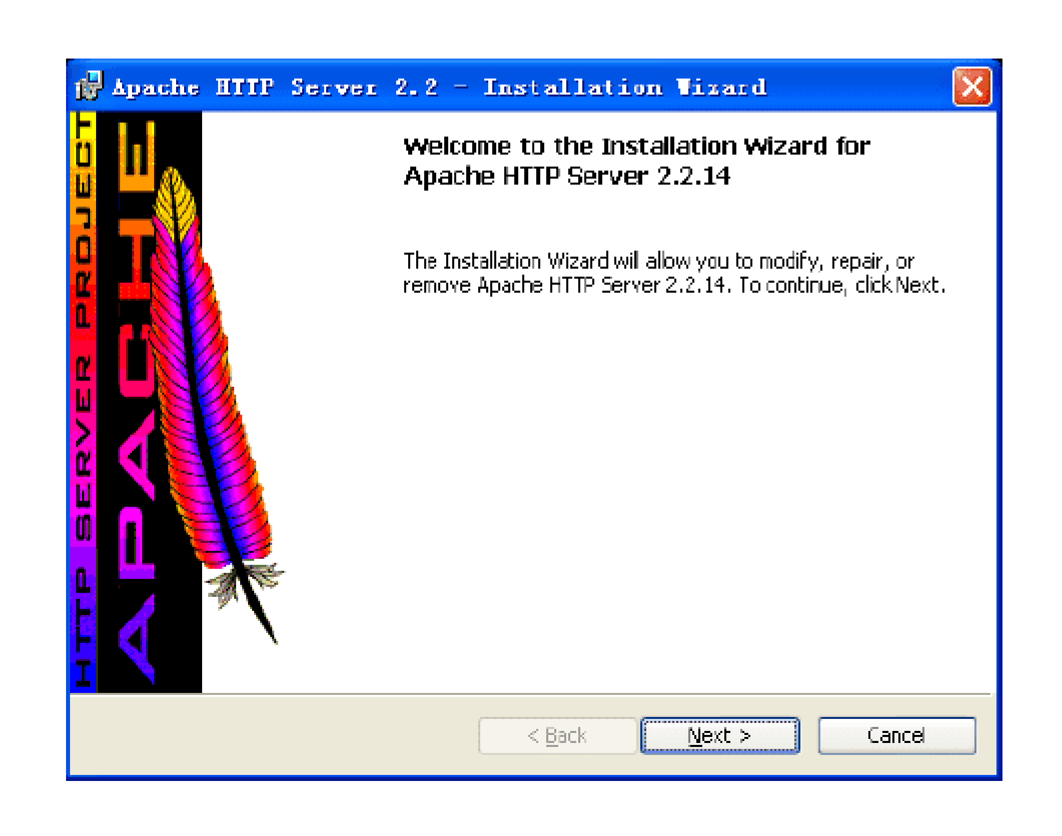


图 1-4　Apache 服务器欢迎安装对话框

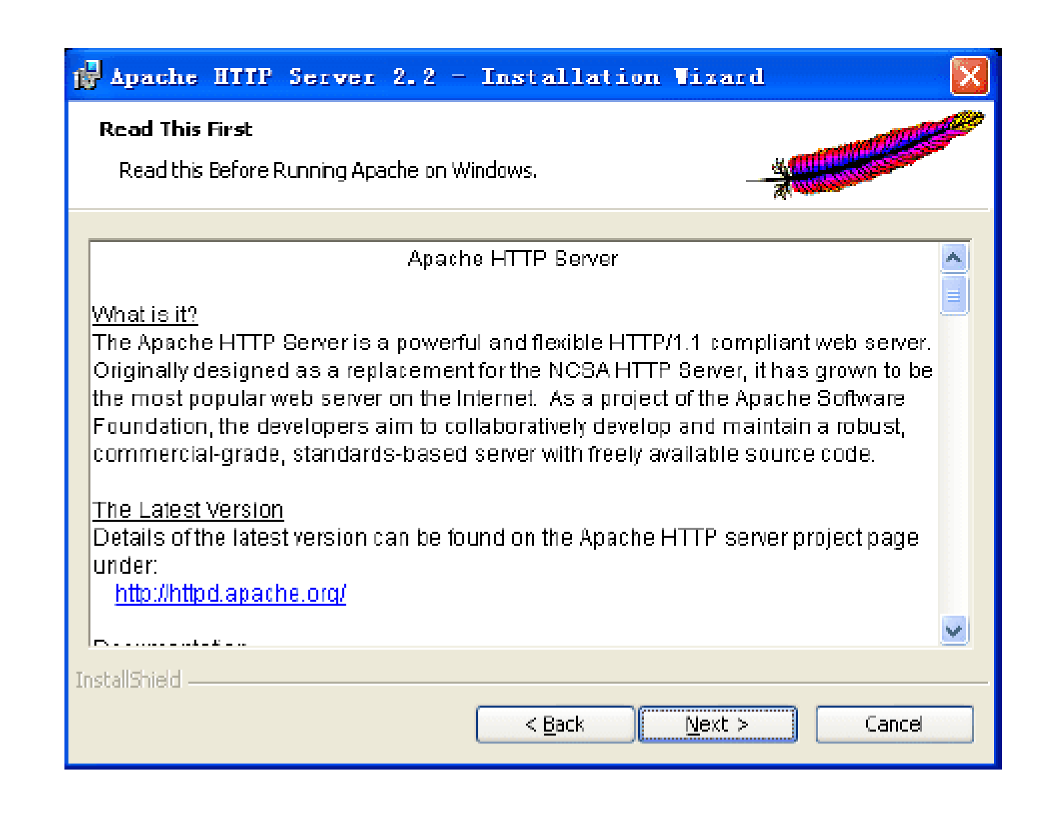


图 1-5　Apache 服务器介绍对话框

(5)　单击图 1-5 中的 Next 按钮，进入 Server Information 向导对话框，如图 1-6 所示。在这个对话框中有三个文本框：第一个文本框要求输入计算机的网络域，默认值为本机的 IP 地址；第二个文本框要求输入服务器的名称，这里输入为 localhost；第三个文本框要求输入系统管理员的电子邮件地址，可以先输入自己的邮件地址。下面的单选按钮是提示希望为所有用户提供 Apache 服务，还是为当前用户提供服务，选中 for All Users,on Port 80,as a Service -- Recommended 单选按钮后，单击 Next 按钮，进入图 1-7 所示的对话框，在这个对话框中选择要安装的类型，有两种类型可供选择，分别是典型安装(Typical)和定制安装(Custom)，这里选择典型安装。

(6)　单击图 1-7 中的 Next 按钮，接下来就进入了如图 1-8 所示的对话框，在这个对话框里可以设定 Apache 的安装目录，单击 Change 按钮，将安装目录设定为 D:\Program Files\Apache Software Foundation\Apache2.2\，如图 1-9 所示。

(7)　设置好安装路径后，单击图 1-8 中的 Next 按钮，进入如图 1-10 所示的对话框，在这

个对话框中，单击 Install 按钮，开始安装 Apache 服务器，如图 1-11 所示。

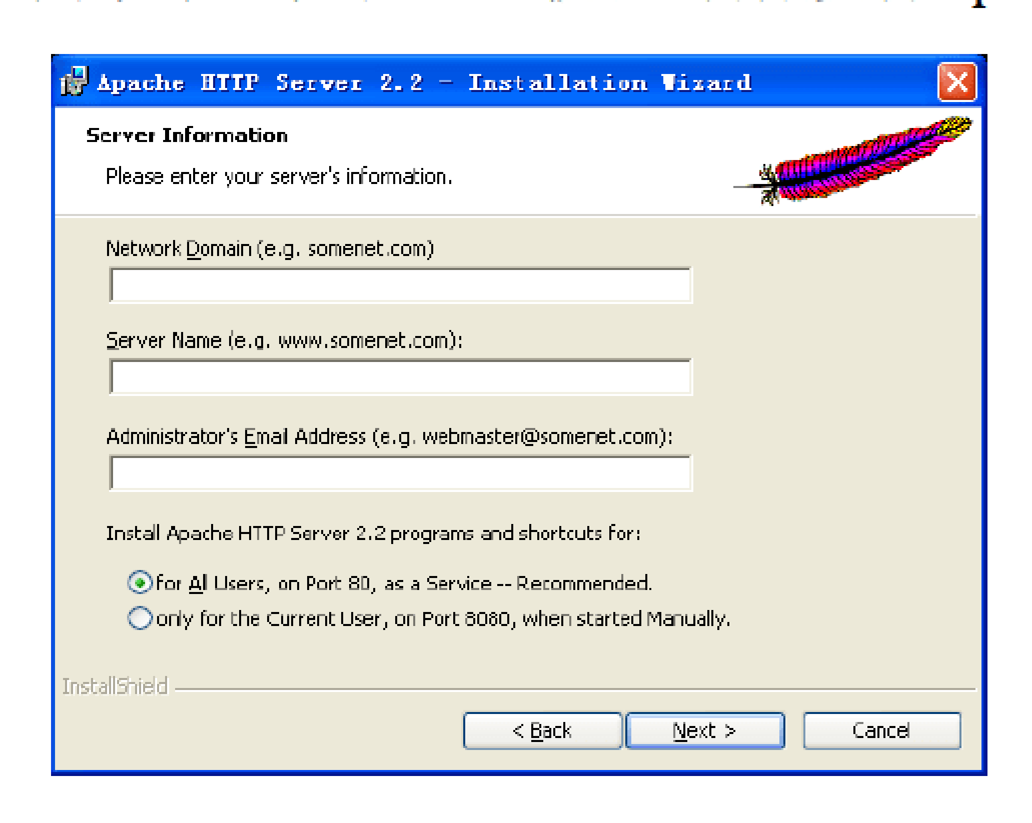


图 1-6　Server Information 对话框

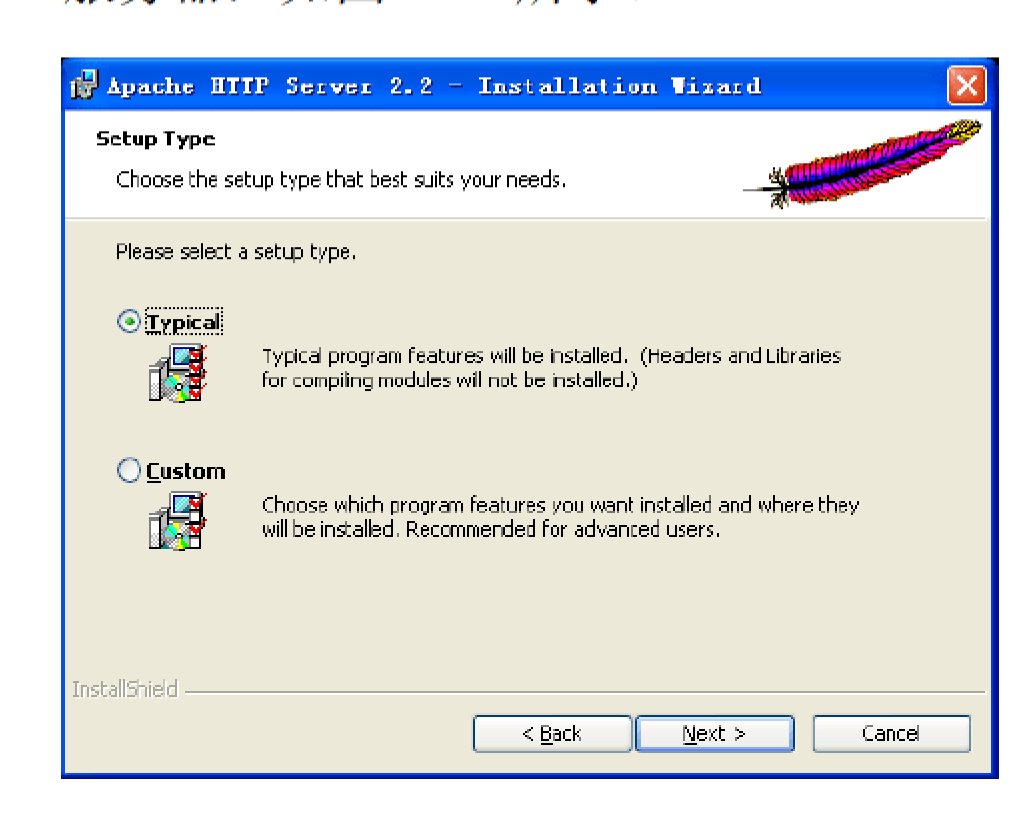


图 1-7　安装类型对话框

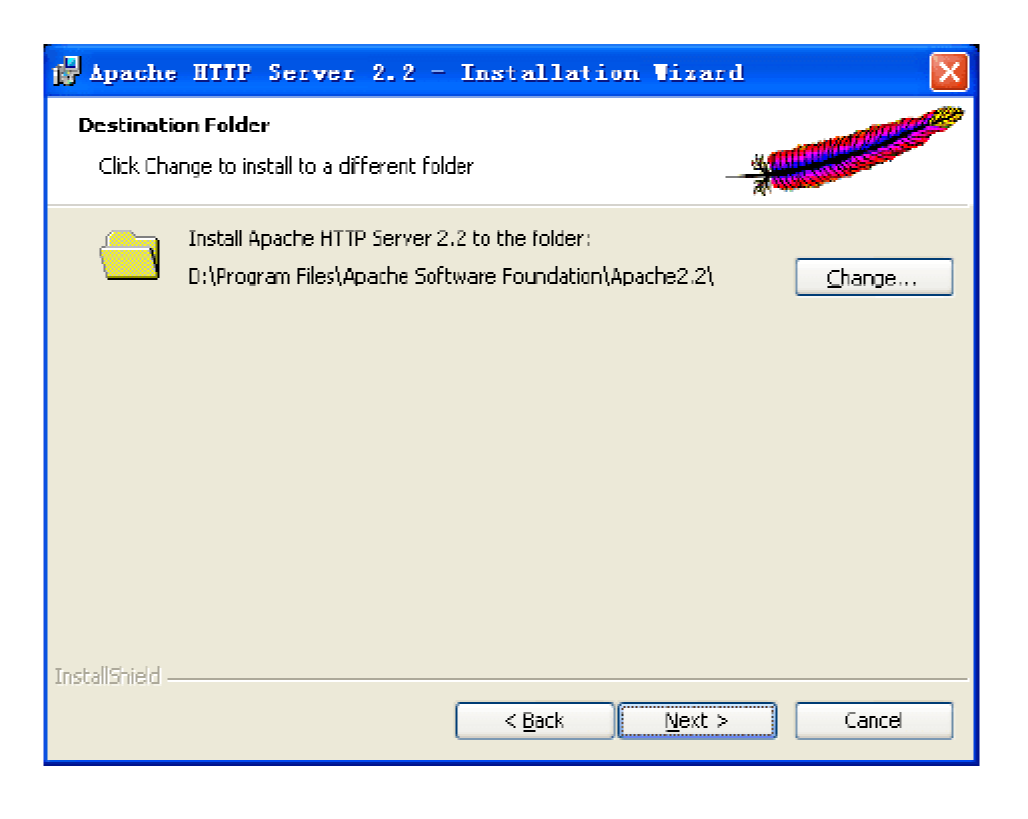


图 1-8　安装路径

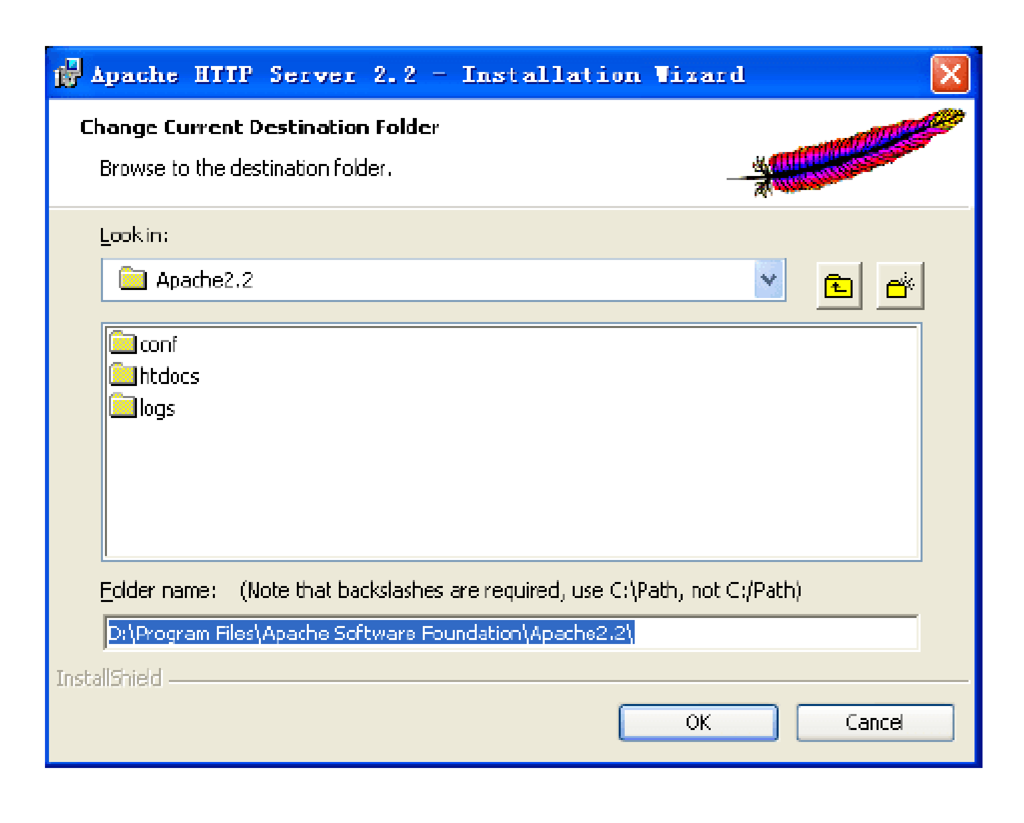


图 1-9　安装路径修改

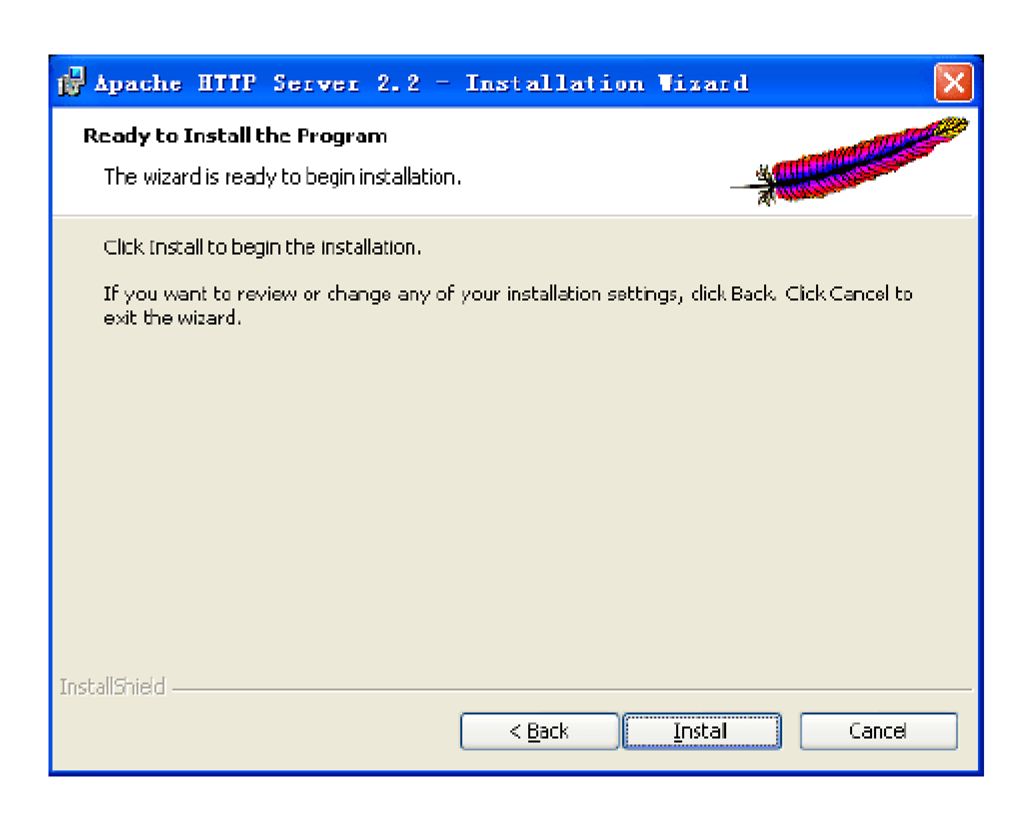


图 1-10　准备安装对话框

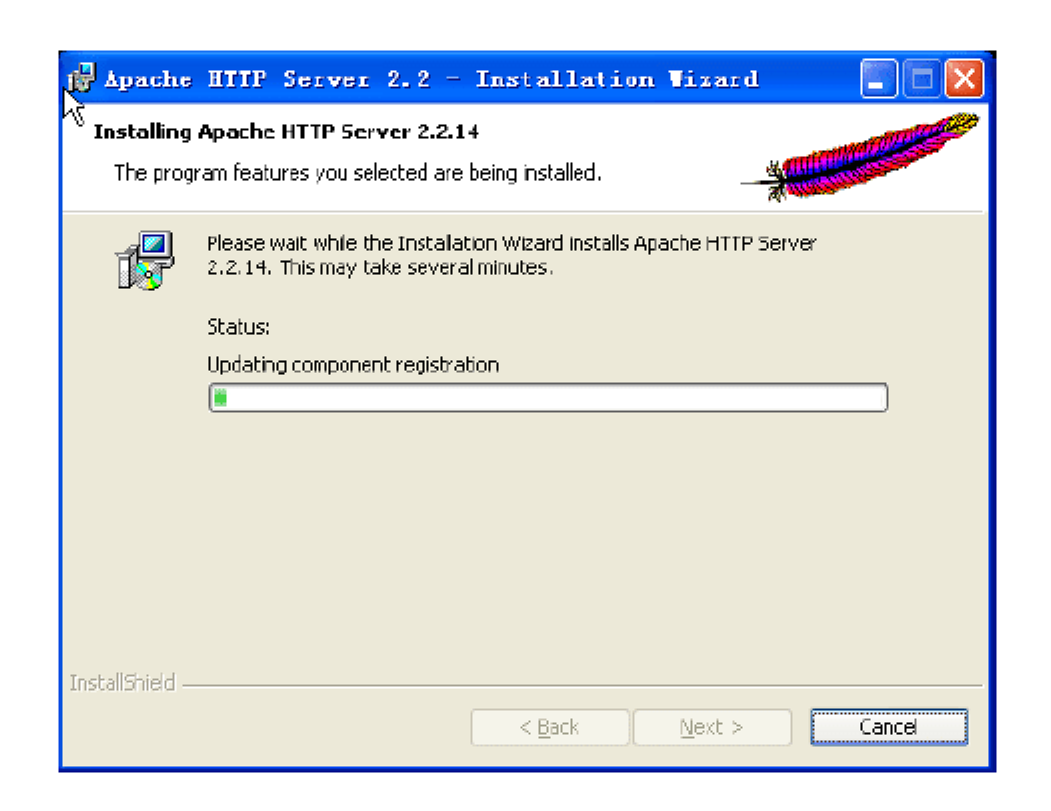


图 1-11　正在安装 Apache

(8) 安装完成后，会出现如图 1-12 所示的对话框，此对话框提示成功安装了 Apache 服务器。

单击 Finish 按钮，完成对 Apache 的安装，在右下角的状态栏会看到图标。下面测试 Apache 服务器是否安装成功和正常启动，打开浏览器，在地址栏中输入 http://localhost/，单击

【转到】按钮，会显示如图 1-13 所示的对话框。

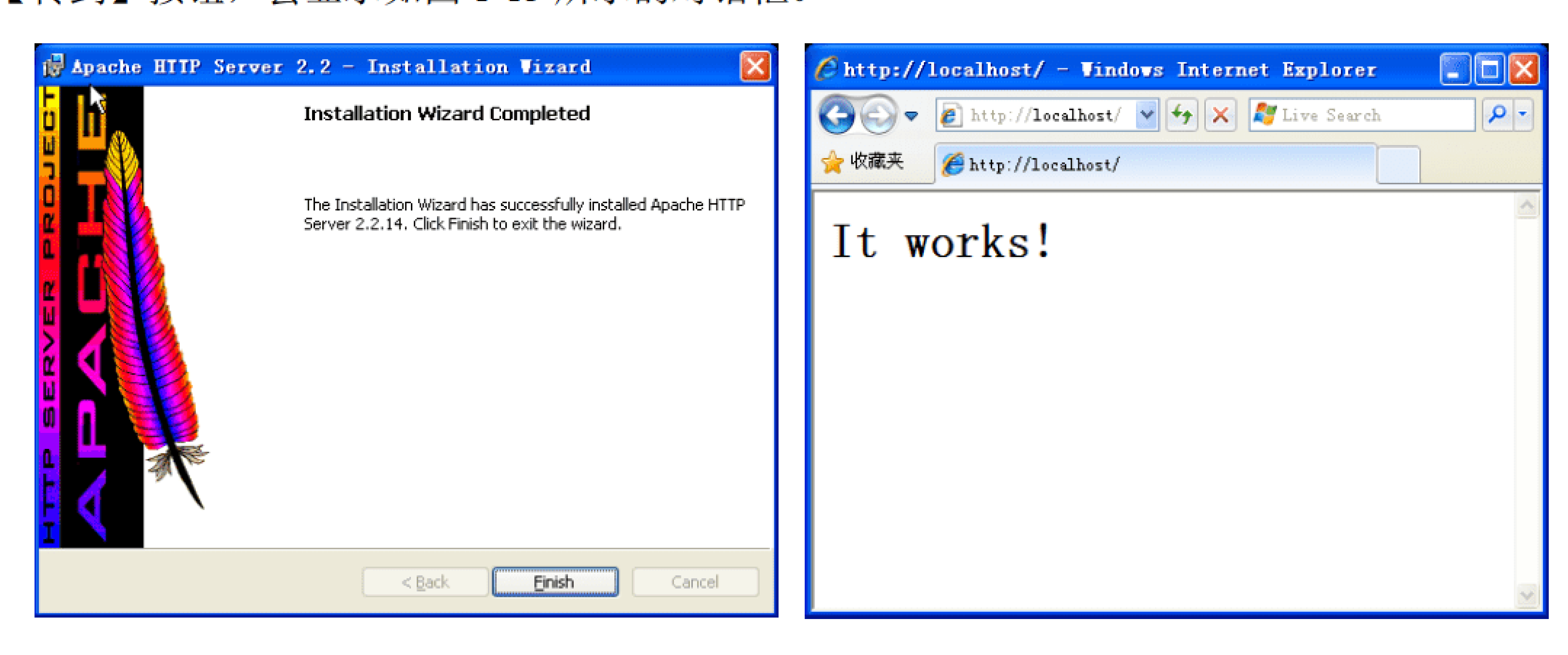


图 1-12　成功安装　　　　图 1-13　Apache 服务器成功启动

1.3.2　如何安装 PHP 环境

开发 PHP 程序时，PHP 开发工具包是核心，只有安装了该工具包，才能解释执行 PHP 页面的脚本程序。下面我们来看一下具体的安装步骤。

(1) 开发工具包要从 PHP 官方网站上下载，打开 IE 浏览器，在地址栏中输入 http://www.php.net/打开网站主页，页面效果如图 1-14 所示。

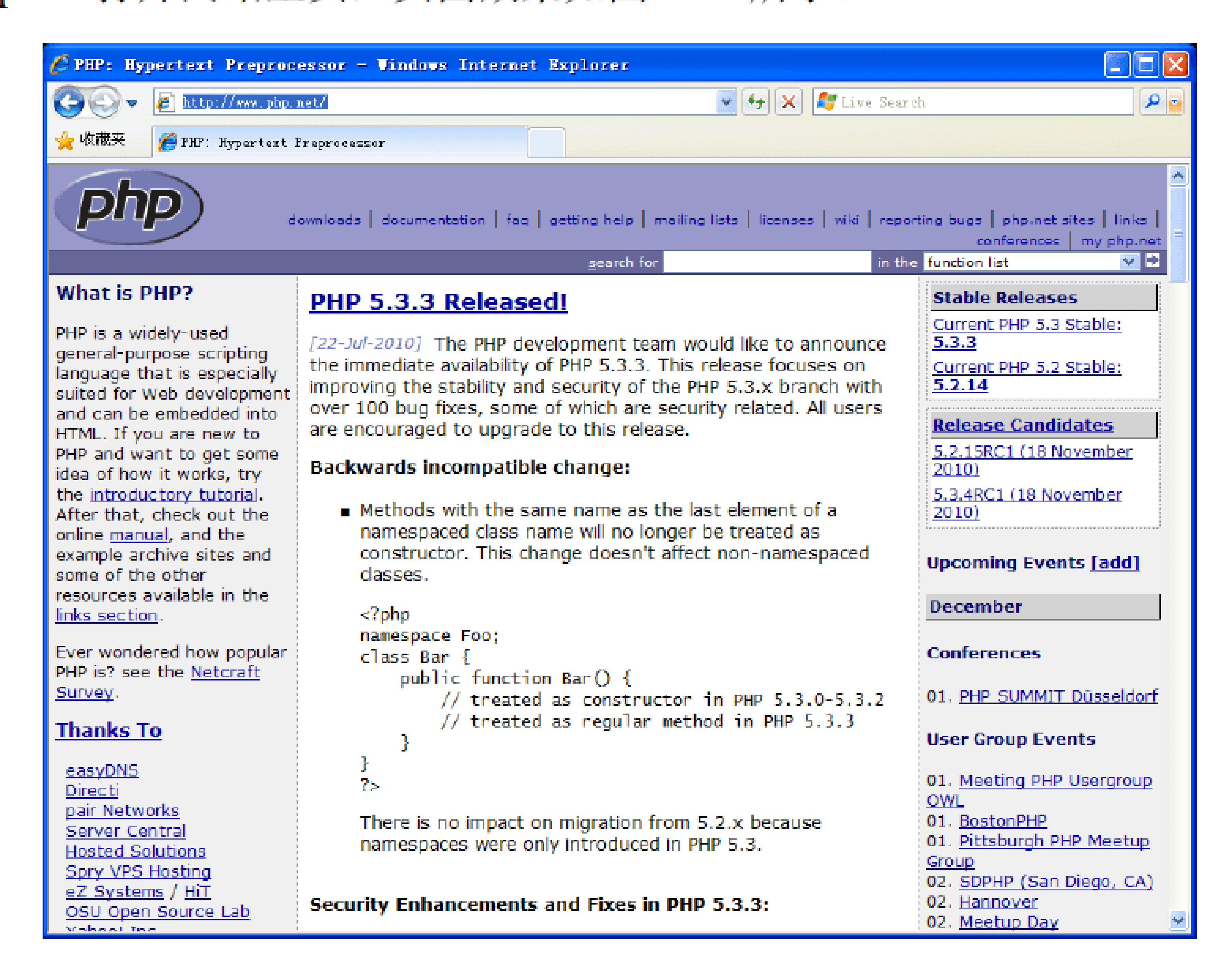


图 1-14　PHP 官方网站

(2) 单击 downloads 超链接，进入 PHP 下载页面。在这里我们可以下载 PHP 工具包，也可以下载 PHP 二进制安装程序，本文主要讲解如何使用 PHP 的安装程序，所以找到能在

Windows 下运行的二进制可执行文件，如图 1-15 所示。单击下载链接后，转到如图 1-16 所示的页面。



图 1-15　PHP 工具包下载信息页面

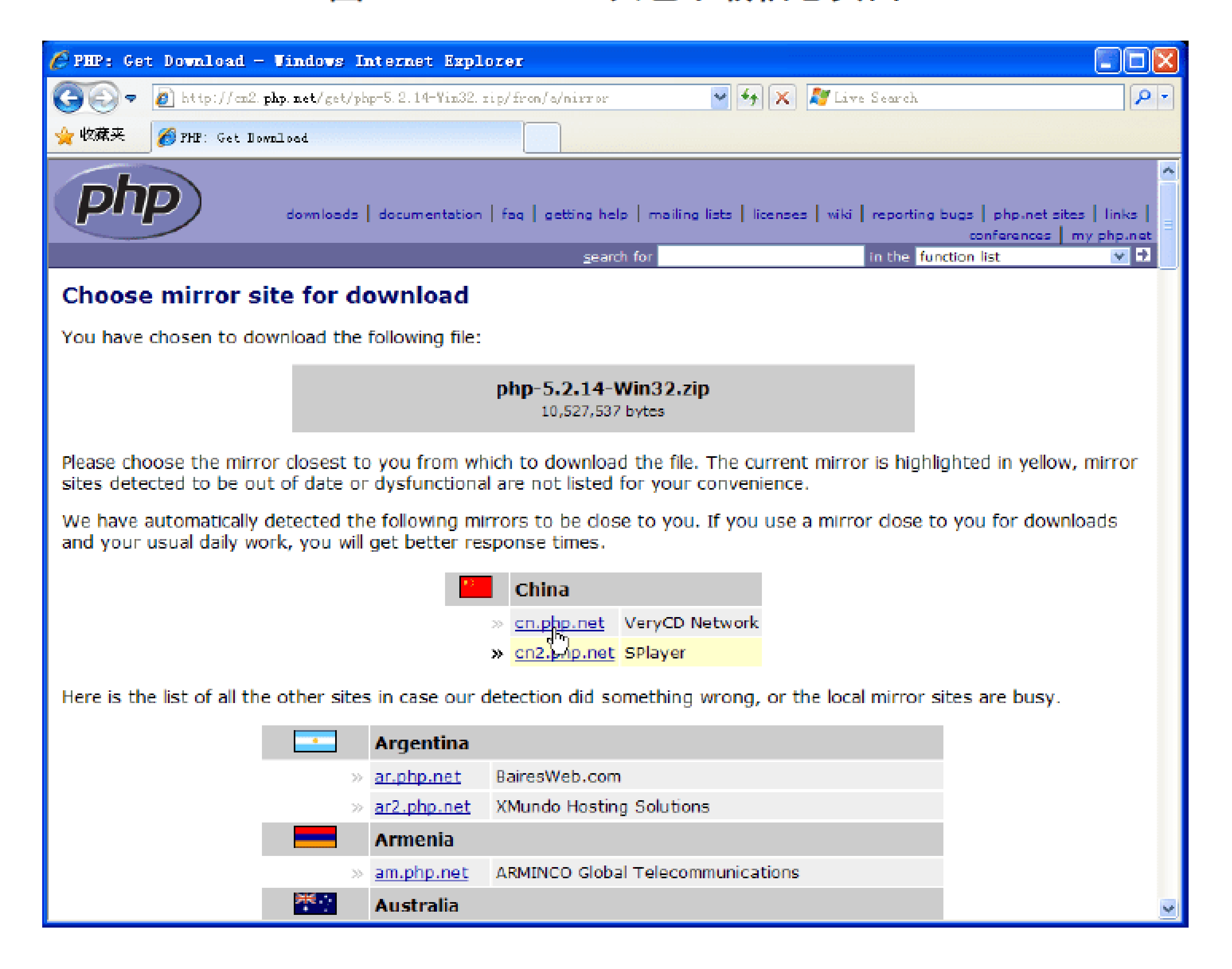


图 1-16　下载链接页面

(3) 下载完成后，双击下载的安装程序，会弹出 PHP 安装程序的安装向导对话框，如图 1-17 所示。

(4) 单击 Next 按钮，弹出 PHP 用户使用协议对话框，如图 1-18 所示。在该用户协议中选中 I accept the terms in the License Agreement 复选框，然后单击 Next 按钮，进入 PHP 工具包安装路径选择对话框，这里我们选择默认路径，如图 1-19 所示。



图 1-17　PHP 安装程序安装向导对话框

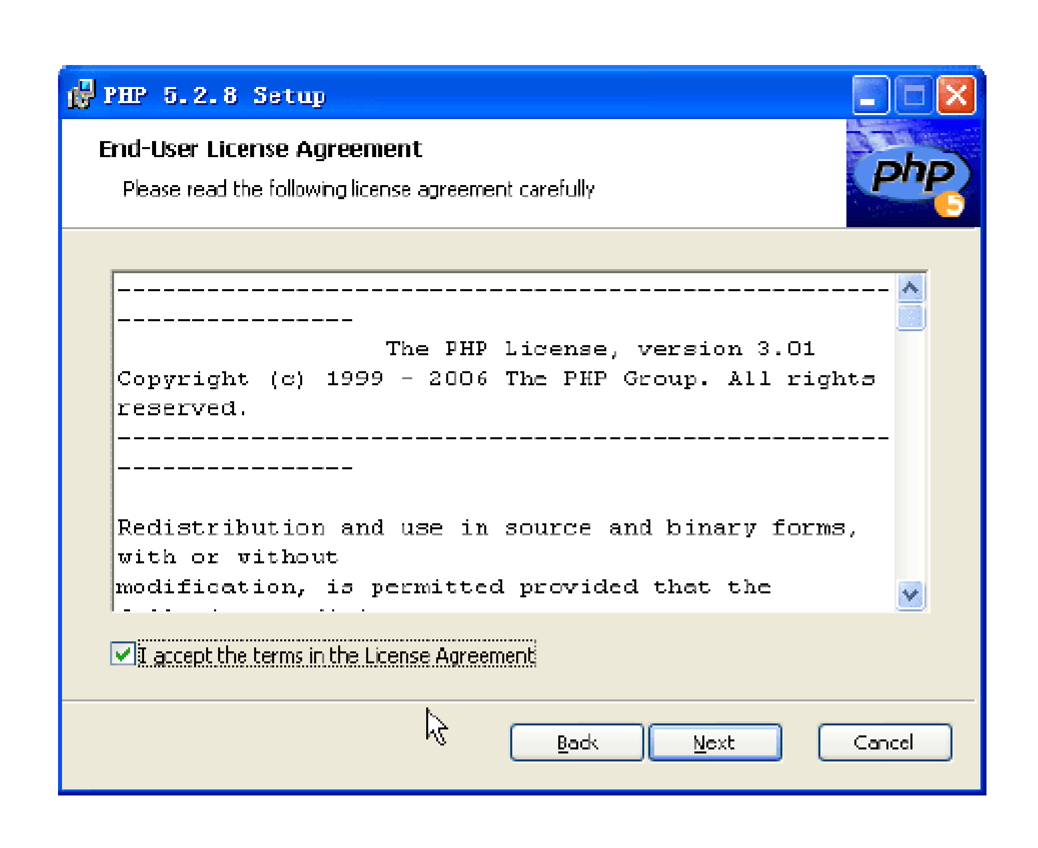


图 1-18　用户使用协议对话框

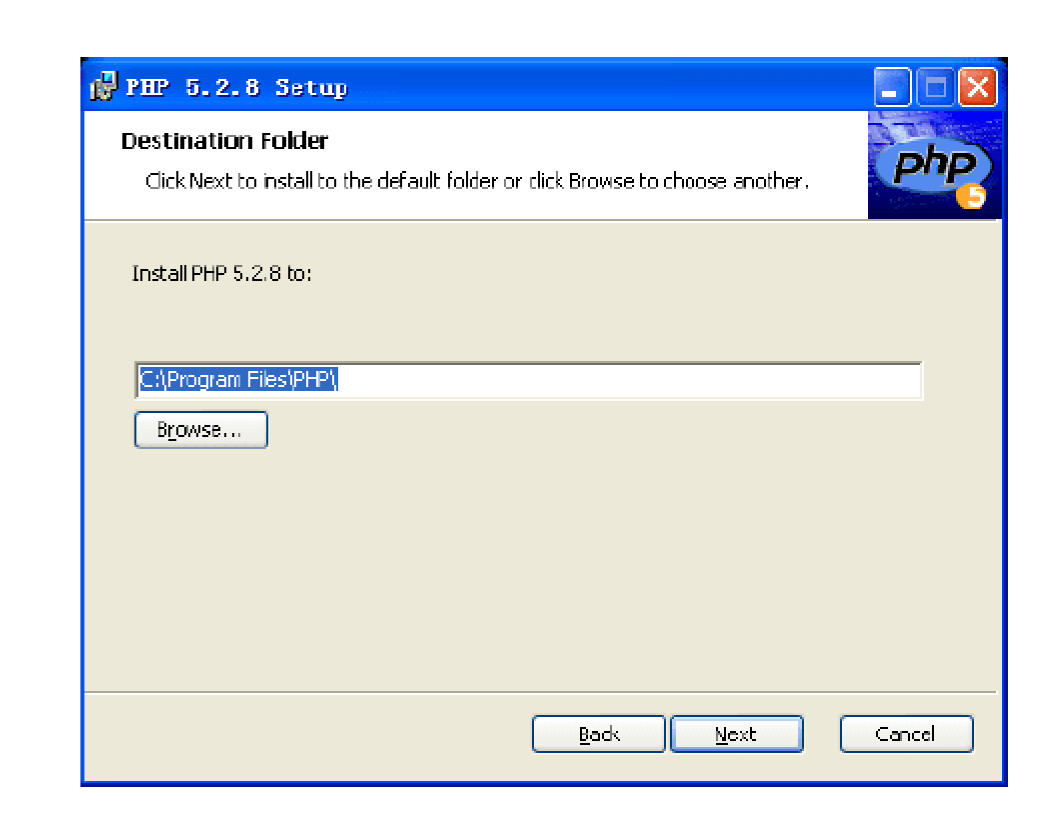


图 1-19　安装路径选择对话框

(5) 安装路径选择好后，单击 Next 按钮，会进入选择 Apache Web 服务器的窗口，这里可以选择要安装的 Apache 服务器的版本，我们选中 Apache 2.2.x Module 单选按钮，如图 1-20 所示。

(6) 选择好 Apache 服务器的版本后，单击 Next 按钮，进入 Apache 服务器的安装路径窗口，如图 1-21 所示。

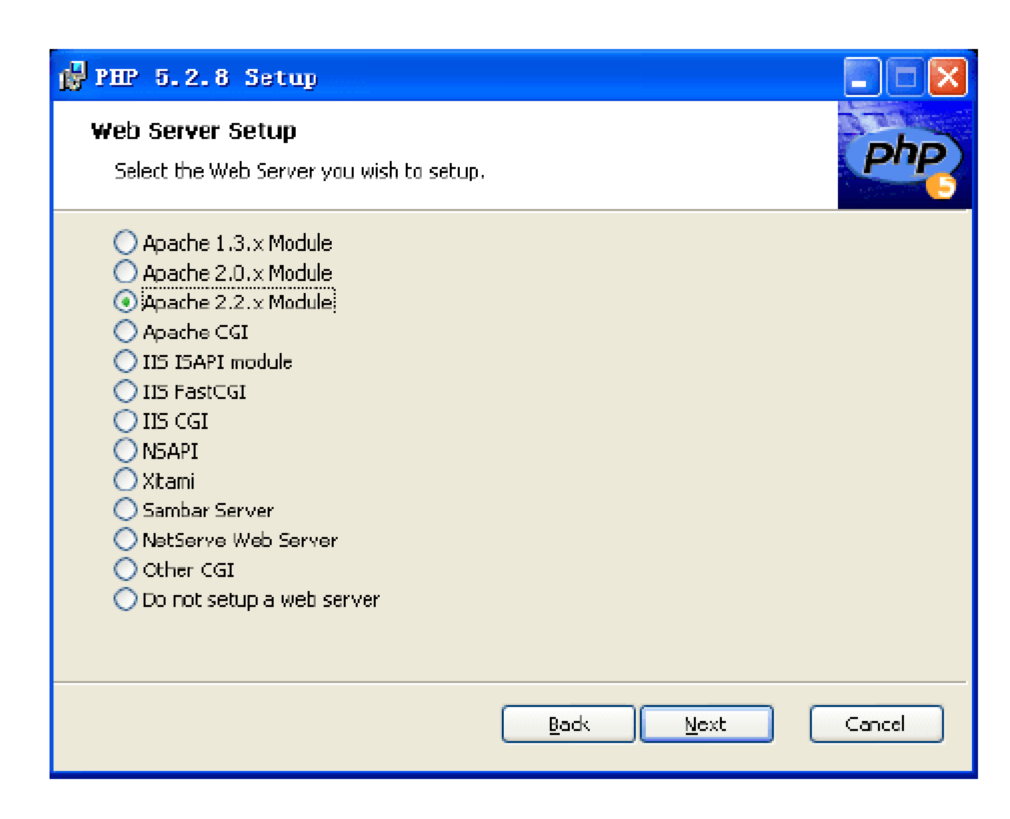


图 1-20　安装 Apache 服务器版本窗口

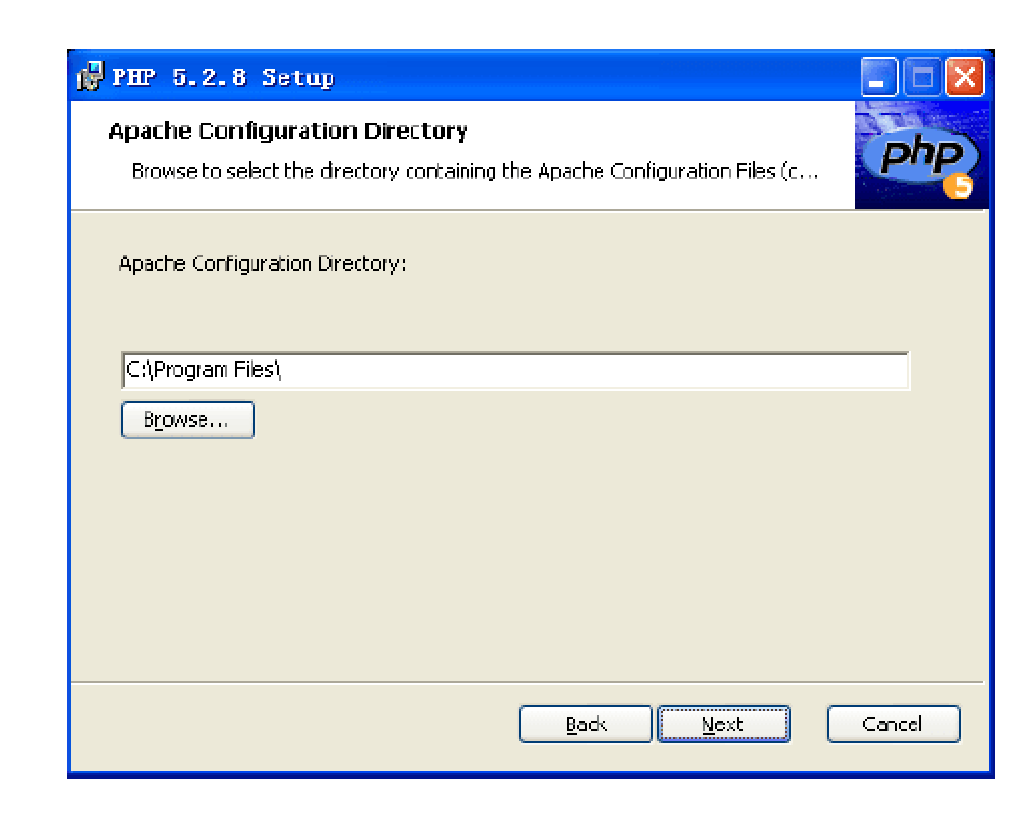


图 1-21　Apache 服务器安装路径窗口

(7) 单击 Browse 按钮，把安装路径更改为我们 1.3.1 小节讲到的 Apache 服务器的安装路径 D:\Program Files\Apache Software Foundation\Apache2.2\，如图 1-22 所示。选择好安装路径

后，如图 1-23 所示，单击 Next 按钮，会进入 PHP 组件安装窗口，在这里我们选择安装所有组件，如图 1-24 所示。

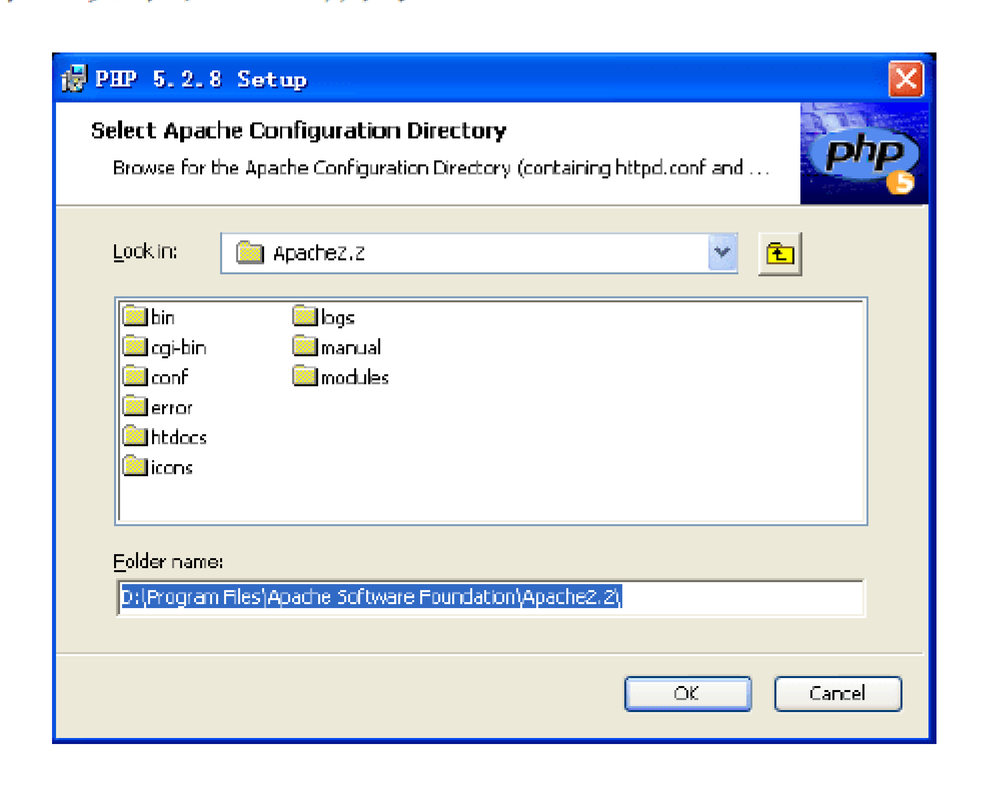


图 1-22　更改安装路径

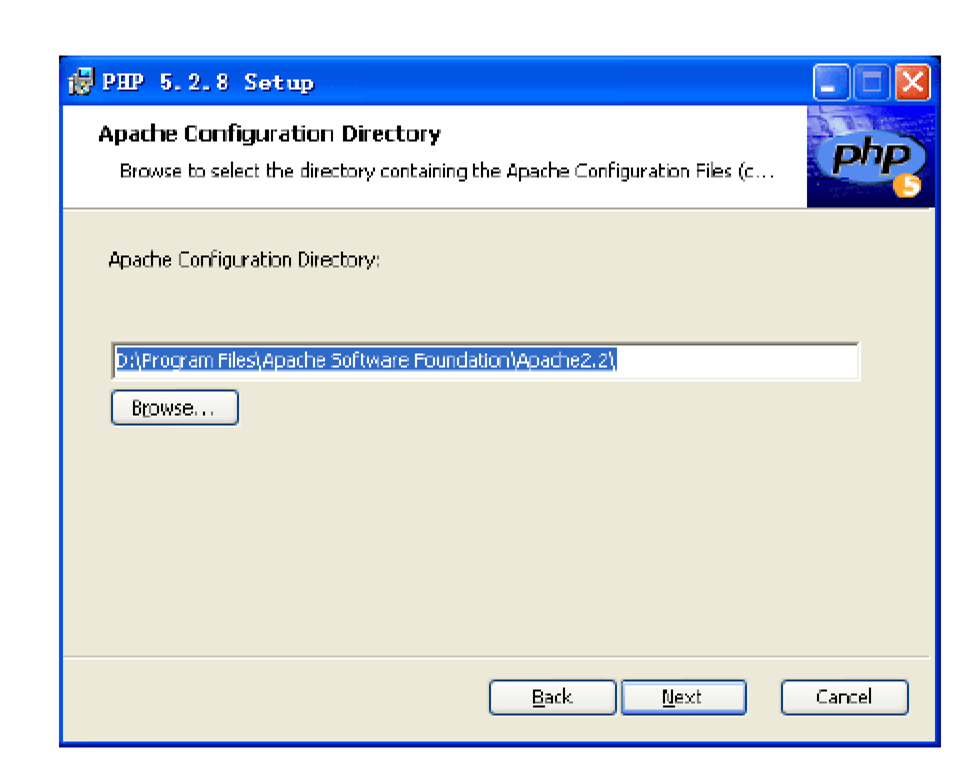


图 1-23　修改后的路径窗口

(8) 在图 1-24 中选择好组件后，单击 Next 按钮，进入 PHP 工具包安装窗口，单击 Install 按钮进行安装，如图 1-25 所示。

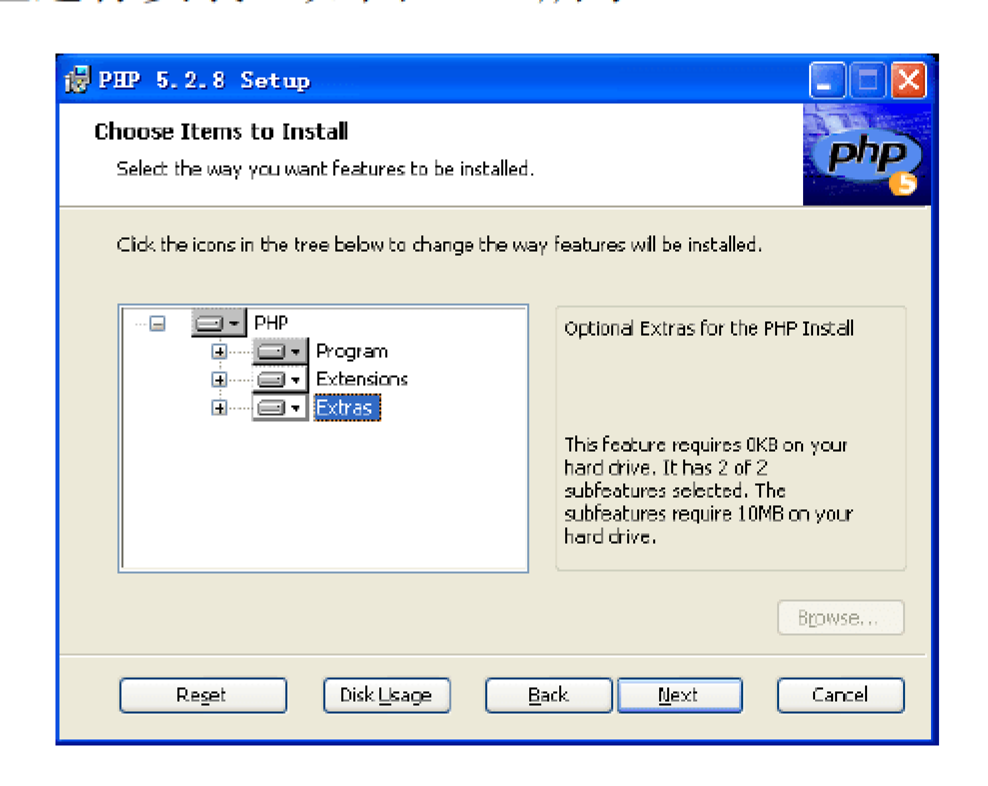


图 1-24　选择组件窗口

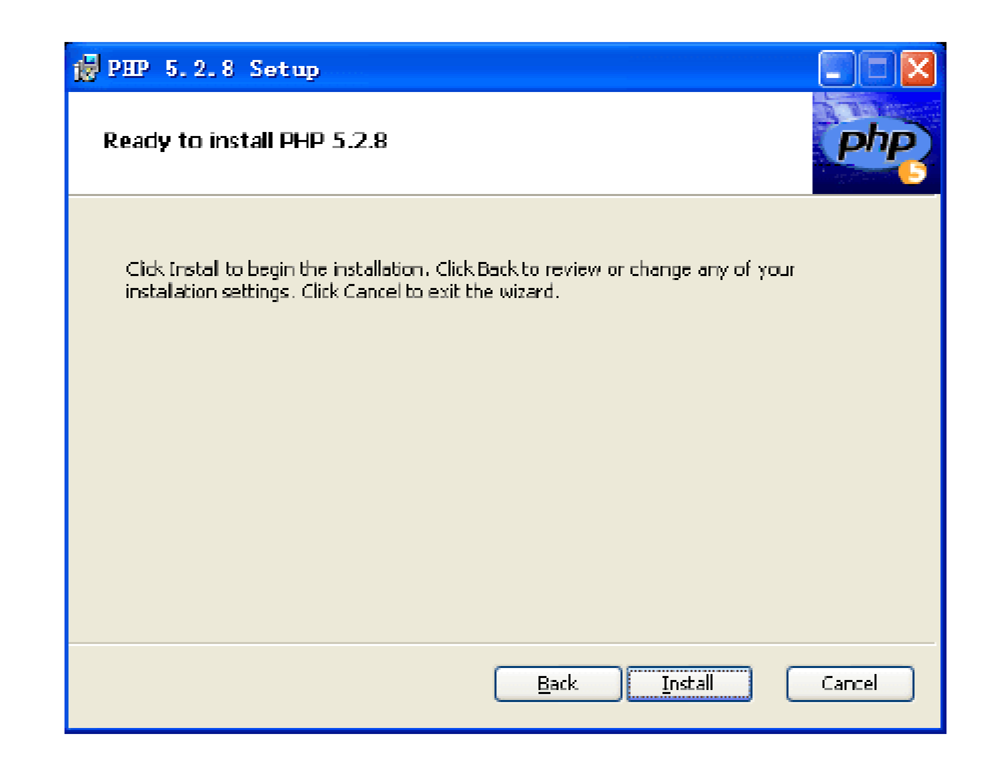


图 1-25　PHP 安装窗口

(9) 程序安装完成后，会显示如图 1-26 所示的程序安装成功的窗口，最后单击 Finish 按钮完成安装。



图 1-26　完成安装

1.3.3　配置与测试 PHP 环境

1. 配置 PHP 环境

PHP 环境的配置需要用到两个重要的文件，分别是 Apache 中的 httpd.conf 和 PHP 中的 php.ini 配置文件。

首先，我们进行 Apache 服务器的配置，在文件夹 D:\Program Files\Apache Software Foundation\Apache2.2\conf 中找到 httpd.conf 文件并打开，找到关键字 DocumentRoot(网站根目录)和 Directory，在这里可以修改成自己的网站目录，此处修改成 D:/www，这样我们就把访问路径配置好了，如图 1-27 所示。

接下来，我们进行 PHP 的配置，在 PHP 的安装目录文件夹 C:\Program Files\PHP 下找到 php.ini 文件，打开配置文件，找到关键字 register_globals，这个值用来打开全局变量，如果其值是 off，就只能用“$_POST[‘变量名’]、$_GET[‘变量名’]”的方式来获取表单中的元素值比较安全，不会让人从网络上轻易截取在网页间传递的数据；如果设为 on，就可以直接用“$变量名”来获取表单中的元素值了，存在安全隐患，如图 1-28 所示。

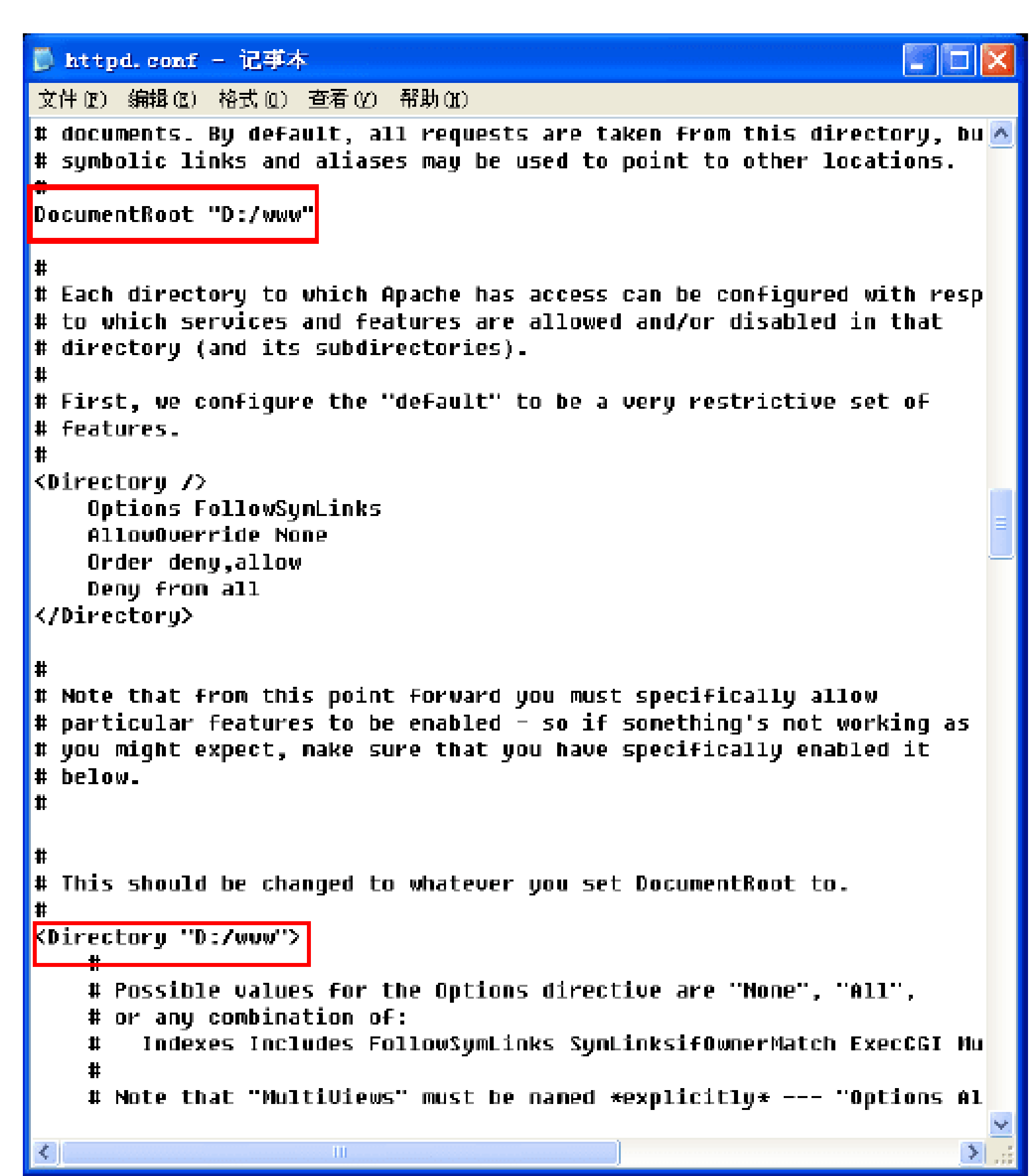


图 1-27　修改网站目录

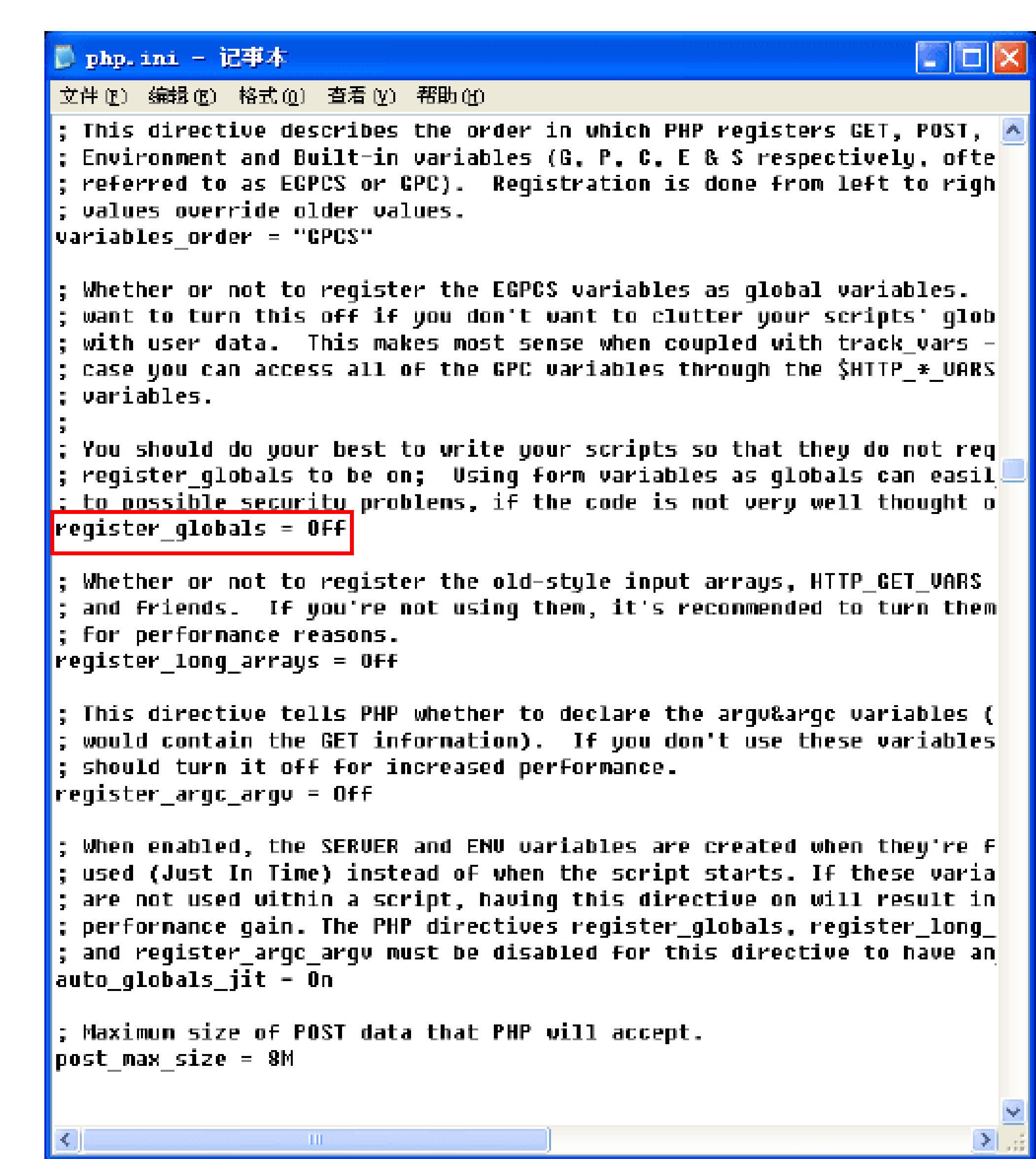


图 1-28　全局变量配置

2. 测试 PHP 环境

前几小节讲解了 Apache 服务器和 PHP 的安装和配置，在这一小节里我们将创建一个 PHP 实例来测试 PHP 工具和 Apache 服务器是否安装成功。如果实例能运行成功，则证明安装成功，否则，安装失败。当然，PHP 和 Apache 的配置我们也将在这一节里具体介绍。

下面就让我们一起做这个小实例，来检验劳动成果吧。

首先测试一下 Apache 服务器是否能够正常启动运行，打开浏览器，在地址栏里输入 http:localhost/，单击【转到】按钮，如果出现如图 1-29 所示的结果，则证明 Apache 服务器能

够正常启动运行。

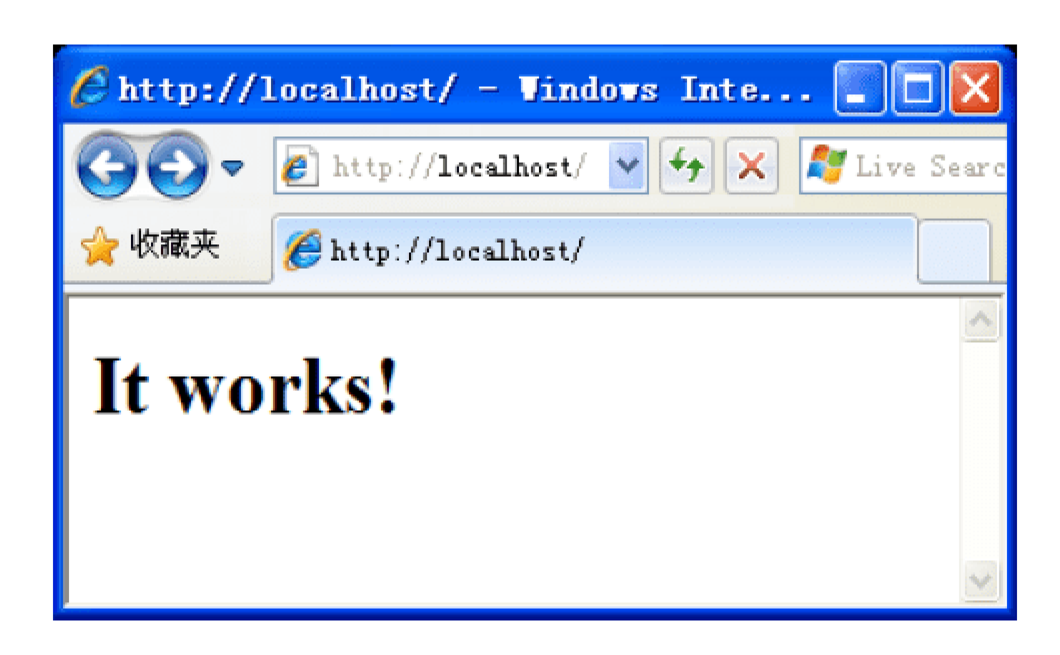


图 1-29　Apache 服务器正常运行

然后找到目录 D:\Program Files\Apache Software Foundation\Apache2.2\htdocs，在目录 htdocs 文件夹里新建一个记事本文档，重命名为 phpinfo.php，在该文件夹中写入如下代码：

```
<?php
  phpinfo();
?>
```

打开 IE 浏览器，在地址栏中输入 http://localhost/phpinfo.php，单击【转到】按钮，如果出现如图 1-30 所示的页面，则证明 PHP 工具包和环境配置成功，该页面显示了 PHP 相关的详细信息。

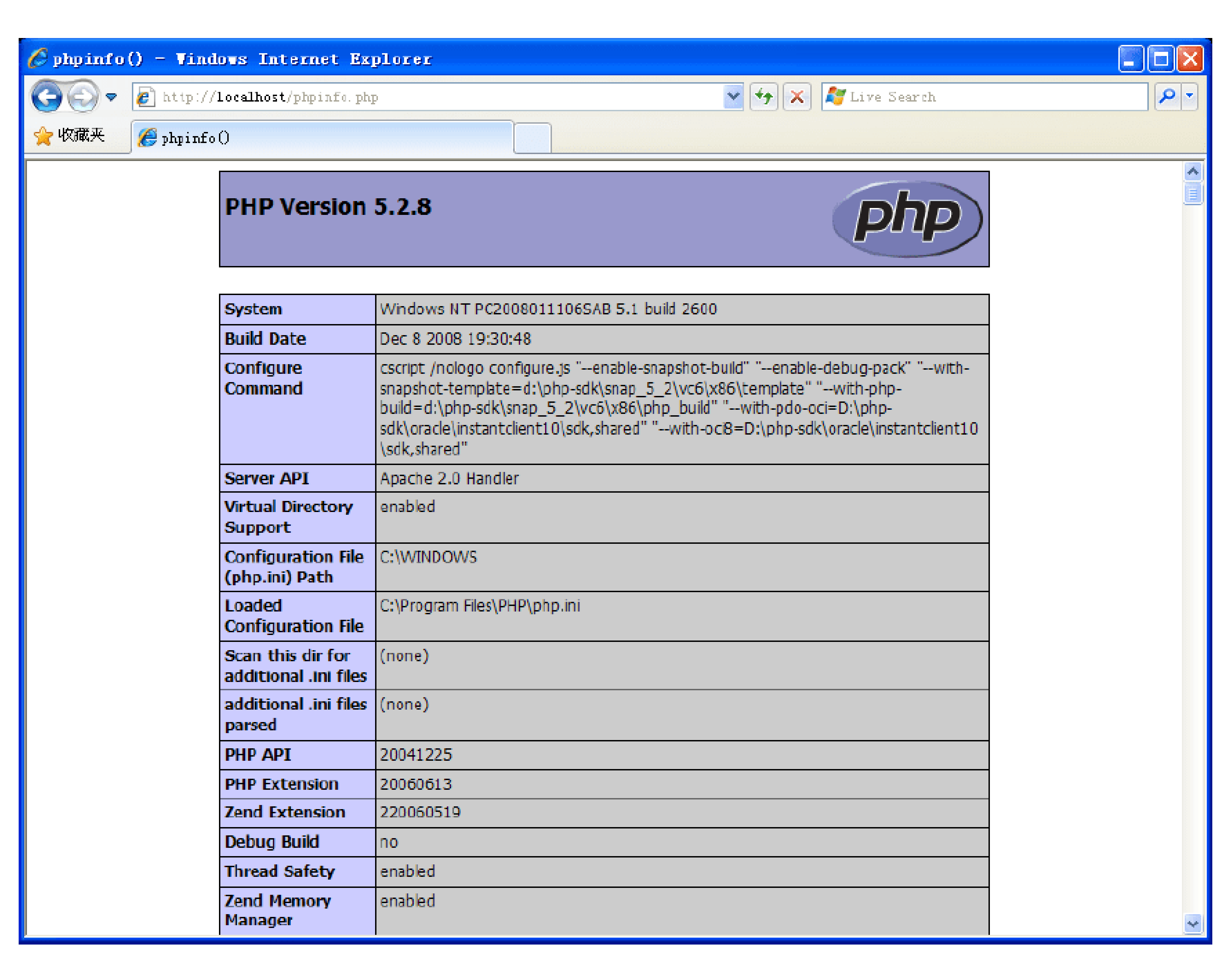


图 1-30　phpinfo.php 的页面



phpinfo()函数提供了 PHP 安装的相关信息，以及组件的配置信息。

1.4　PHP 配置指令

根据 PHP 的特性可以了解到 PHP 功能的强大，但通过对下面 PHP 配置指令的了解，你会逐渐地意识到 PHP 不仅拥有强大的功能体系，而且使用起来更方便。

视频教学：光盘\videos\02\设置 PHP 扩展.avi　　长度：4 分钟
光盘\videos\02\PHP 配置指令.avi　　长度：6 分钟
光盘\videos\02\管理配置指令.avi　　长度：6 分钟

Apache 的 httpd.conf 文件和 PHP 的 php.ini 文件包含了大量的配置指令，这些配置指令控制了 Apache 和 PHP 的行为，并决定了 PHP 的运行效果，这里我们详细讲解 PHP 常见的配置指令。

1.4.1　管理配置指令

1. php.ini

php.ini-recommended 和 php.ini-dist 是 PHP 的两个配置模板，前者中的参数设置都已经设为了推荐值，如果使用前者，在调整安装时就能节省大量的时间和精力，因为这个文件中的大约 240 个参数的默认值可以使你很快地部署 PHP。我们下面就来学习和了解文件中的配置参数。

php.ini 文件是 PHP 全局配置文件，它处理了 PHP 在 12 个不同方面的行为，如表 1-3 所示。

表 1-3　PHP 的不同方面的 12 种行为

语言选项	数据处理
安全模式	路径和目录
语法突出显示	文件上传
杂项	Fopen 包装器
资源限制	动态扩展
错误处理和日志	模块设置
语言选项	数据处理

php.ini 是一个纯文本文件，只包含注释和“参数=值”赋值对，代码如下：

```
;;;;;;;;;;;;;;;;;
; Data Handling ;
;;;;;;;;;;;;;;;;;
;
; Note - track_vars is ALWAYS enabled as of PHP 4.0.3

; The separator used in PHP generated URLs to separate arguments.
; Default is "&".
;arg separator.output = "&amp;"
```

```
; List of separator(s) used by PHP to parse input URLs into variables.
; Default is "&".
; NOTE: Every character in this directive is considered as separator!
;arg separator.input = ";&"

; This directive describes the order in which PHP registers GET, POST, Cookie,
; Environment and Built-in variables (G, P, C, E & S respectively, often
; referred to as EGPCS or GPC). Registration is done from left to right, newer
; values override older values.
variables order = "GPCS"

; Whether or not to register the EGPCS variables as global variables. You may
; want to turn this off if you don't want to clutter your scripts' global scope
; with user data. This makes most sense when coupled with track_vars - in which
; case you can access all of the GPC variables through the $HTTP * VARS[],
; variables.
;
; You should do your best to write your scripts so that they do not require
; register globals to be on; Using form variables as globals can easily lead
; to possible security problems, if the code is not very well thought of.
register globals = Off

; Whether or not to register the old-style input arrays, HTTP_GET_VARS
; and friends. If you're not using them, it's recommended to turn them off,
; for performance reasons.
register_long_arrays = Off
```

在以上代码中，以分号开头的行是注释或者说明，如果清楚某个参数配置的作用，可以直接删掉分号，启用该配置，这样就大大简化了编辑时间。

修改过的配置，如果安装 PHP 作为二进制包，那么在每次调用 PHP 时都会重启 php.ini 文件，且立即生效。但如果 PHP 作为 Apache 的一个模块，那么要在 Apache 服务器重新启动后，修改的配置才能生效。

2. Apache 的 httpd.conf 和.htaccess 配置文件

当 PHP 作为 Apache 的模块运行时，我们需要通过 httpd.conf 或者.htaccess 配置文件修改指令，可以使用“指令二值”形式来设置，如表 1-4 所示为最常用的指令。

表 1-4　指令表

指　令	说　明
php_value	设置指定指令的值
php_admin_value	设置指定指令的值，但不能在.htaccess 文件中设置或覆盖
php_flag	设置指定布尔指令的值
php_admin_flag	设置指定布尔指令的值，但不能在.htaccess 文件中设置或覆盖

3. 在脚本中执行

通过 ini_set()方法，可以用最本地化的方式处理 PHP 配置变量，如要修改 PHP 中给定脚本的最大执行时间，只需加入以下命令即可。

```
int_set("max_execution_time","60");
```

4. 指令的作用域

PHP 设置了 4 个指令作用域，考虑到安全，我们修改指令时，都只能在指令自己的作用域中进行修改，如表 1-5 所示。

表 1-5　指令作用域

名　称	作　用
PHP_INI_PERDER	指令可以在 php.ini、httpd.conf 和.htaccess 文件中进行修改
PHP_INI_SYSTEM	指令可以在 php.ini 和.htaccess 文件中进行修改
PHP_INI_USER	指令可以在用户脚本中进行修改
PHP_INI_ALL	指令可以在任何地方进行修改

1.4.2　PHP 的配置指令

相信通过 1.4.1 节的学习，应该已经知道怎样对 PHP 配置指令进行配置和管理了，在这一节里，将对 php.ini 文件里比较常用的配置指令进行详细说明。

1. 语言选项

语言选项指令主要确定语言最基本的行为，其指令的详细配置信息如表 1-6 所示。

表 1-6　语言选项配置指令

指　令	作 用 域	详细信息
engine	PHP_INI_ALL	确定 PHP 引擎是否可用
zend.zel_compatibility_mode()	PHP_INI_ALL	启用该指令，可以是 PHP 5 和 PHP 4 兼容
short_open_tag	PHP_INI_ALL	允许使用段标记
asp_tags	PHP_INI_ALL	是否使用 ASP 风格的脚本界定符
precision(integer)	PHP_INI_ALL	指定浮点数中显示有效数字的个数
output_buffering()	PHP_INI_SYSTEM	是否使用缓冲
output_hander()	PHP_INI_ALL	确定在输出返回之前是否要将所有输出传递给一个函数

2. 语法突出显示

PHP 可以突出显示源代码。通过将 PHP 脚本的扩展名更改为.phps，或者利用 show_source()方法或 higeliget_file()函数就可以启用此功能，如果要使用.phps 扩展名，需要向 httpd.conf 中增加以下代码：

```
AddType application/x-httpd-php-source .phps
```

对于突出显示的源码，可以通过以下 6 个指令控制字符串、注释、关键字、背景、默认文

本和 HTML 的部分颜色。

1) highlight.string(string)

默认值：#DD0000；作用域：PHP_INI_ALL

2) highlight.comment(string)

默认值：#FF9900；作用域：PHP_INI_ALL

3) highlight.keyword(string)

默认值：#007700；作用域：PHP_INI_ALL

4) highlight.bg

默认值：#FFFFFF；作用域：PHP_INI_ALL

5) highlight.default

默认值：#0000BB；作用域：PHP_INI_ALL

6) highlight.html

默认值：#000000；作用域：PHP_INI_ALL

3. 安全模式

我们在多用户系统环境中部署使用 PHP 时，可能要限制 PHP 的许多功能。因为 PHP 在多用户使用时会暴露服务器的漏洞，还有可能破坏服务器资源和文件。所以，为了安全使用 PHP，在部署时可以使用受限模式和安全模式。

但是，启用安全模式会有很多影响，包括自动禁用很多特性和不安全的功能。我们可以使用如表 1-7 的指令进行修改，限制 PHP 指令。

表 1-7 安全模式指令

指 令	作 用 域	详细信息
safe_mode()	PHP_INI_SYSTEM	表示 PHP 在上述条件下运行
safe_mode_gid()	PHP_INI_SYSTEM	启用安全模式，在打开文件时，强制完成 UID 检测
safe_mode_include_dir()	PHP_INI_SYSTEM	打开文件时忽略 UID 检测
safe_mode_exec_dir()	PHP_INI_SYSTEM	限制通过 exec()函数
safe_mode_allowed_envars()	PHP_INI_SYSTEM	限制用户使用 PHP 脚本修改系统变量
safe_mode_protected_envars()	PHP_INI_SYSTEM	限制修改变量
open_basedir(string)	PHP_INI_ALL	新建一个可以操作的目录
ignore_user_abort()	PHP_INI_SYSTEM	使服务器忽略由中断所造成的会话终止
disable_functions(string)	PHP_INI_SYSTEM	禁用函数
disable_class(string)	PHP_INI_SYSTEM	禁用类

4. 资源限制

在处理程序时，为了确保 PHP 脚本不会被独占，可以使用以下 3 个指令来控制脚本的执

行时间、脚本输入处理时间和内存，如表 1-8 所示。

表 1-8　资源限制指令

指　令	作 用 域	详细信息
max_execution_time(integer)	PHP_INI_ALL　默认值为 30	设置脚本执行上限
max_input_time(integer)	PHP_INI_ALL　默认值为 60	设置 PHP 解析请求数据所用的时间
memory_limit(integer)	PHP_INI_ALL　默认值为 8MB	设置一个脚本所申请的最大内存数

5. 数据处理

在 PHP 中，外部变量就是通过一些外部源传递给脚本的变量，例如 POST、GET、cookie、服务器和操作系统提供的外部数据。以下指令可以影响外部变量的使用，如表 1-9 所示。

表 1-9　数据处理命令

指　令	作 用 域	详细信息
register_globals()	PHP_INI_SYSTEM	设置全局变量
register_long_arrays()	PHP_INI_SYSTEM	是否使用已废弃的语法
register_argc_argv()	PHP_INI_SYSTEM	设置通过 GET 方式传入变量，和执行文件传递参数类似
magic_quotes_gpc()	PHP_INI_SYSTEM	是否启用魔法引号
magig_quotes_sybase()	PHP_INI_ALL	在启用魔法引号后，使用单引号进行转义
auto_prepend_file()	PHP_INI_SYSTEM	设置要加载的文件名和路径
default_mimetype()	PHP_INI_ALL	设置 PHP 文件类型
default_charset()	PHP_INI_ALL	设置 PHP 文件首部，输出字符编码格式

6. 文件上传

在 PHP 中，PHP 文件支持 POST 方法上传和管理本文件及二进制文件。我们可以使用以下 3 个指令实现上传功能，如表 1-10 所示。

表 1-10　文件上传

指　令	作 用 域	详细信息
file_uploads()	PHP_INI_SYSTEM	是否允许 HTTP 文件上传
upload_tmp_dir()	PHP_INI_SYSTEM	设置存放文件的临时目录
upload_max_filesize()	PHP_INI_SYSTEM	设置上传文件的最大字数

7. 错误处理和日志

PHP 为报告和记录 PHP 在编译和运行时生成的错误、警告和提示，提供了一种方便灵活的方法。开发人员可以控制报告的敏感度，例如这些信息是否在浏览器中显示及如何显示，信息将记录在文件中和系统日志中等。

在了解指令之前，我们先看看 PHP 的错误报告级别，如下表 1-11 所示。

表 1-11　PHP 的错误报告级别

级 别 名	级别描述
E_ALL	报告所有错误和报告
E_ERROR	报告致命的运行时错误
E_WARNIG	报告非致命的运行时错误
E_PARSE	报告编译时的解析错误
E_MOTICE	报告运行时注意信息，如未初始化变量
E_STRICT	PHP 版本可移植性建议
E_CORE_ERROR	报告在 PHP 启动时发生的致命错误
E_CORE_WARNING	报告在 PHP 启动时发生的非致命性错误
E_COMPILE_ERROR	报告致命的编译时错误
E_COMPILE_WARNING	报告非致命的编译时错误
E_USER_ERROR	报告用户引发的致命错误信息
E_USER_WARNING	报告用户引发的非致命错误信息
E_USER_NOTICE	报告用户引发的注意消息

在应用程序开发和初始部署阶段，应当将敏感度设置为最高级别，如 E_ALL。但是，一旦解决了主要的 bug 之后，就要考虑降低敏感度。

当然，我们可以通过表 1-12 所示的指令控制错误处理行为。

表 1-12　错误处理和日志指令

指　令	作 用 域	默 认 值	详细信息
error_reporting(string)	PHP_INI_ALL;	NULL	确定 PHP 错误报告敏感度的级别
display_errors(on,off)	PHP_INI_ALL	ON	输出 error_reporting 指定级别上的所有错误
display_startup_errors(on,off)	PHP_INI_ALL	OFF	禁用时，可防止将 PHP 启动过程中的错误显示给用户
log_errors(on,off)	PHP_INI_ALL	OFF	启用时，告诉 PHP 应当记录错误信息，可以记录在某个文件中或者 syslog 中
log_errors_max_len(integer)	PHP_INI_ALL	1024	确定一条日志信息的最大长度，以字节为单位。参数设置为 0 时不限制最大长度
ignore_repeated_errors(on,off)	PHP_INI_ALL	OFF	禁用后，防止记录重复错误
ignore_repeated_source(on,off)	PHP_INI_ALL	OFF	禁用后，忽略重复错误时将不考虑错误来源，每个错误消息只能记录一次

续表

指　令	作 用 域	默 认 值	详细信息
repeat_memleaks(on,off)	PHP_INI_ALL	OFF	在 PHP 以调试模式编译时有效，确定是否显示或记录内存泄露
track_errors(on,off)	PHP_INI_ALL	OFF	启用时，会让 PHP 在$php_error_msg 变量中存储最近的错误信息，其作用域仅限于出现错误的特定脚本内
html_errors(on,off)	PHP_INI_SYSTEM	ON	默认时，PHP 把错误信息包括在 HTML 标记内
docref_root(string)	PHP_INI_ALL	NULL	启用时，PHP 会包括一个指向错误详细描述的链接，一般指向官方手册
docref_ext(string)	PHP_INI_ALL	NULL	用于提供关于错误的额外信息时，本参数会告诉 PHP 本地手册页面的扩展名
error_prepend_string(string)	PHP_INI_ALL	NULL	在输出错误之前向用户传递附加信息
error_append_string(string)	PHP_INI_ALL	NULL	在输出错误之后向用户传递附加信息
error_log(string)	PHP_INI_ALL	NULL	启用时将指定消息目标，PHP 支持将错误记录在指定文件或系统日志中。在 Windows 下，将本参数设置为 syslog 会把消息记录在事件日志中

8. 路径和目录

PHP 默认的路径设置指令是用于导入函数库和扩展包，以及确定用户 Web 目录和 Web 文档根目录的，如表 1-13 所示。

表 1-13　路径和目录指令

指　令	作 用 域	详细信息
include_path(string)	PHP_INI_ALL	此参数指定的路径是 include()、require()和 fopen_with_path()等函数使用的基本路径
doc_root(string)	PHP_INI_SYSTEM	此参数提供所有 PHP 脚本的默认位置
user_dir(string)	PHP_INI_SYSTEM	指定在使用/~username 约定打开文件时，PHP 所使用的绝对目录
extension_dir(string)	PHP_INI_SYSTEM	告诉 PHP 加载的扩展包(模块)的位置，在默认情况下，其值为./，表示可加载的扩展包与所执行的脚本位于相同的位置
enable_dl(on,off)	PHP_INI_SYSTEM	允许用户在运行时加载 PHP 扩展包，即在脚本执行期间加载

9. fopen 包装器

fopen 包装器指令与访问和处理远程文件有关，如表 1-14 所示。

表 1-14　fopen 包装器

指　令	作 用 域	详细信息
allow_url_fopen(on,off)	PHP_INI_ALL	启用后，使 PHP 可以将远程文件看作本地文件，如果有远程文件的权限，PHP 脚本能够访问和修改这些文件
from(string)	PHP_INI_ALL	确定用于完成 FTP 连接的匿名用户密码
user_agent(string)	PHP_INI_ALL	确定用户带来属性
Default_scoket_timeout(integer)	PHP_INI_ALL	确定基于套接字的流的超时值，以秒为单位
auto_detect_line_endings(on,off)	PHP_INI_ALL	启用时，将确定 fgets()和 file()读取的数据使用的是 Macintosh、MS-DOS 还是 UNIX 的文件约定

10. mail

在 PHP 中，mail()函数为通过 PHP 脚本发送电子邮件提供了一个很方便的途径。PHP 有 4 个指令来确定它的这种行为，如表 1-15 所示。

表 1-15　Mail 指令

指　令	作 用 域	详细信息
SMTP(string)	PHP_INI_ALL	默认值：localhost，只用于 Win32 操作系统，确定了发送邮件时 PHP 所要用的 SMTP 服务器的 DNS 名或 IP 地址
smtp_port(int)	PHP_INI_ALL	默认值：25，只用于 Win32 操作系统，指定了 PHP 通过服务器发送邮件时所用的端口
sendmail_from(string)	PHP_INI_ALL	默认值：null，只用于 Win32 操作系统，指定了 PHP 开始发送邮件时所用的发送标识
sendmail_path(string)	PHP_INI_SYSTEM	默认值：DEFAULT_SENDMAIL_PATH，只用于 UNIX 操作系统，用来向 sendmail 守护进程传递加载项

11. Java

PHP 还可以和 Java 程序一起使用，通过其 Java 扩展包来实例化 Java 类，表 1-16 所示的 4 个指令确定了 PHP 的这种行为。PHP 也可以通过 Java servlet API 把自己作为一个 Java servlet 运行。

表 1-16　Java 指令

指　令	作 用 域	详细信息
java.class.path(string)	PHP_INI_ALL	默认值：null，java.class.path 目录指定 Java 类存储的位置

续表

指　令	作 用 域	详细信息
java.home(string)	PHP_INI_ALL	默认值：null，指定 JDK 二进制文件的目录位置
java.library(string)	PHP_INI_ALL	默认值：null，指定 Java 虚拟机(JVM)的位置
java.library.path(string)	PHP_INI_ALL	默认值：null，指定 PHP 的 Java 扩展部的位置

1.5　常见问题解答

1.5.1　处理 GET 和 POST 请求方法问题



处理 GET 和 POST 请求方法问题？
网络课堂：http://bbs.itzcn.com/thread-13631-1-1.html

PHP 是否仅限于处理 GET 和 POST 请求方法呢？

【解决办法】：不是，PHP 有可能处理任何请求方法，例如 CONNECT。适当的回应状态可以用 header() 发送。如果仅需要处理 GET 和 POST 方法，可以通过如下的 Apache 配置实现：

```
<LimitExcept GET POST>
Deny from all
</LimitExcept>
```

1.5.2　php.ini 的路径问题



php.ini 的路径问题？
网络课堂：http://bbs.itzcn.com/thread-13632-1-1.html

在 Windows 系统下，我已经安装了 PHP，但在哪可以找到 php.ini 配置文件呢？

【解决办法】：Windows 中 php.ini 文件的默认路径在 Windows 目录下。如果使用的是 Apache 服务器，则会首先在 Apache 的安装目录中寻找 php.ini，例如 C:\ProgramFiles\ApacheGroup\Apache。这样，同一台机器上不同版本的 Apache 就可以有不同的 php.ini 文件。

1.6　习　　题

一、填空题

(1) PHP 是__________的简称。
(2) PHP5 在__________年成为了 PHP 唯一的维护中的稳定版本。
(3) 数据处理指令 register_globals()的作用是__________。

二、选择题

(1) 以下不是 PHP 语言的特点的是__________。

A. 开放性源代码　　B. 运行于服务器端

C. 跨平台性　　D. 面向过程性

(2) 在 PHP 配置指令的语言选项中，用__________指令确定 PHP 引擎是否可用。

A. short_open_tag　　B. asp_tags

C. precision(integer)　　D. zend.zel_compatibility_mode()

(3) 以下__________指令不是安全模式中的指令。

A. safe_mode_gid()

B. safe_mode_allowed_envars()

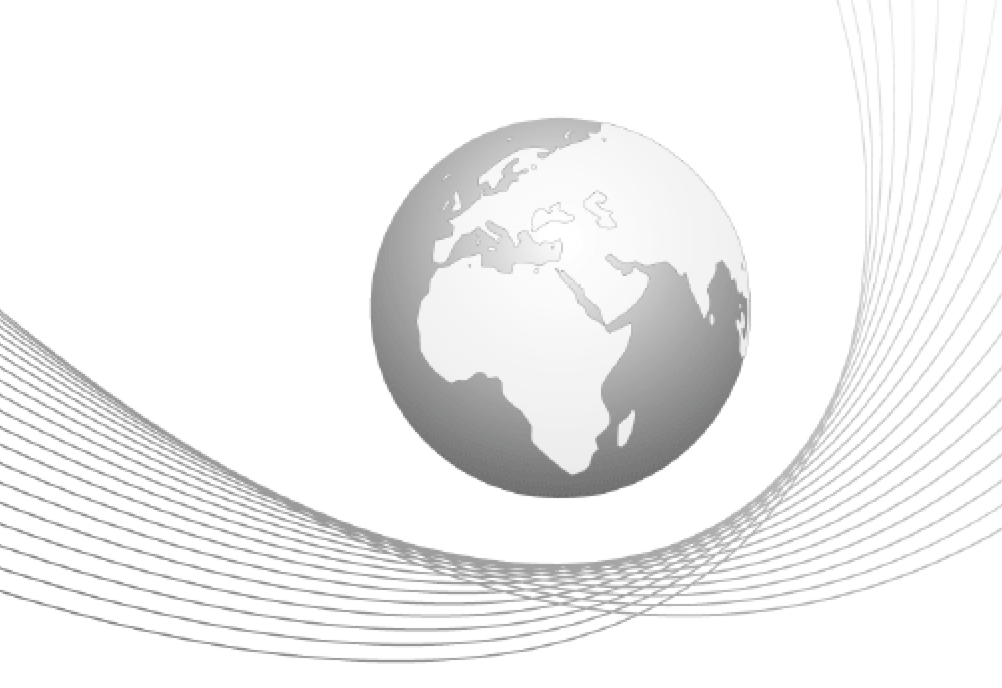
C. log_errors(on,off)

D. safe_mode_include_dir()

三、上机练习

上机练习： 自己动手安装和配置 Apache 服务器和 PHP。

练习要求：能够手动安装 Apache 服务器和 PHP，并使用 safe_mode()指令启用 PHP 的安全模式，使用 safe_mode_include_dir()指令忽略对打开文件的 UID 检测。



第 2 章　PHP 基础语法

内容摘要：

PHP 语言跟微软的 ASP 有几分相似，都是在服务器端执行应嵌入到 HTML 文档中的脚本语言。PHP 语言具有很独特的语法，对初学者来说学起来比较容易。本章将详细地介绍 PHP 语言的众多基础语法。

学习目标：

- 掌握如何在 HTML 中嵌入 PHP 代码。
- 了解 PHP 的各种注释方法。
- 掌握 PHP 的各种数据类型的使用以及类型之间的转换。
- 掌握 PHP 变量的使用。
- 了解 PHP 常量的使用。
- 熟练掌握 PHP 表达式以及操作数和操作符。
- 掌握 PHP 的控制结构。
- 熟练掌握 PHP 的函数。
- 熟悉 PHP 的数组。

2.1 界定 PHP 代码

PHP 是一个可以直接嵌套在 HTML 语言中的脚本语言。如果将嵌入 PHP 的页面全部由 PHP 引擎进行解释，效率会很低下。这样解析器就需要一种方法来确定页面中哪部分是 PHP 代码，并只针对这部分代码进行解析，从而提高效率。

这种方法就是将 PHP 脚本语言放置到特定的标记内，这种标记可以被解析器识别，告诉解析器 PHP 脚本语言的开始和停止位置，而此标记外的内容将被 PHP 解析器忽略。

 **视频教学：**光盘\videos\02\ PHP 基本语法.avi　　长度：14 分钟

2.1.1 基础知识——嵌入 PHP 代码

大多数情况下，PHP 脚本语言都是嵌入在 HTML 文档中的，在 HTML 文档中使用 PHP 脚本语言常用的界定形式有 4 种，下面将逐一进行介绍。

1. 默认语法

默认的界定语法是以“<?php”为开始标记，以“?>”为结束标记。如以下代码：

```
<?php
  echo "php 嵌套默认语法";
?>
```

代码运行时，PHP 解析器只解析开始标记“<?php”和结束标记“?>”中的代码，即使用 echo 输出函数输出“php 嵌套默认语法”。

2. 短标记

PHP 中还有更短的界定语法，称为短标记。在这种标记中省略了默认语法中的 PHP 引用，要使用这个特性，必须启用 PHP 配置指令中的 short_open_tag 指令，把它的值设为 on，这种标记才能使用。如以下代码：

```
<?
  echo "这是短标记语法";
?>
```

3. 脚本标记

PHP 在被某些编辑器处理时，有时会出现语法上的问题。因此，PHP 还支持一种主流的界定标记，这种脚本界定标记是以“<script language="php">”标记开始，以“</script>”标记结束，在其中嵌入 PHP 代码，这样也可以把 PHP 语言与其他语言界定开。如以下代码：

```
<script language="php">
  echo "这是脚本标记语法";
</script>
```



微软的 FrontPage 编辑器也可以识别 ASP 风格的 PHP 语法界定。

4．ASP 风格

Microsoft ASP 页面使用了一种类似的策略，使用预定义的字符模式将静态内容和动态内容分开，以<%开头，以%>结束。该标记中可以放置 PHP 代码，使用方式和 ASP、JSP 中嵌入脚本程序的标记一样。例如：

```
<%
   print "ASP 风格标记";
%>
```

2.1.2　实例描述

PHP 是嵌入式语言，在一个页面中，PHP 代码和非 PHP 代码可以多次交替出现，上面所讲的几种界定语法也可以混合使用。虽说我们可以使用多种标记界定 PHP 代码，但还是要以最基本的默认语法为主，这样可以避免某些编辑器不识别的问题。在以下的实例中我们将详细地讲解各种界定的使用。

2.1.3　实例应用

【例 2-1】 嵌入多个代码块

(1) 新建一个名为 2.1.1.php 的文件，在文件中的<title>标签内我们使用 PHP 默认标记，代码如下所示：

```
<title><?php echo "使用多个代码块"?></title>
```

(2) 在以下代码中，我们也使用默认标记定义了$str 变量。

```
<h1>使用多个代码块</h1>
  <?php
  $str="这里是窗内网,欢迎光临！！！";
  $date="2010-12-17"
  ?>
```

(3) 接下来定义一个<a>标签，在<a>标签中使用短标记输出在第(2)步中定义的变量。代码如下所示：

```
<a href="http://www.itzcn.com">
    <h1><font color="#FF0000">
       <?=$str;?>
    </font></h1>
  </a>
```

(4) 定义一个<h2>标签，在这个标签中使用 PHP 脚本标记输出一段文字，代码如下所示：

```
<h2>
```

```
    <script language="php">
      echo "我们使用了 PHP 的多个代码块<br>";
    </script>
  </h2>
```

(5) 保存这个文件，我们就写好了一个可以嵌套 PHP 语言的文件了。

2.1.4 运行结果

将这个页面放在 Apache 服务器中，启动 Apache 服务器，在浏览器中打开，将显示如图 2-1 所示的效果。

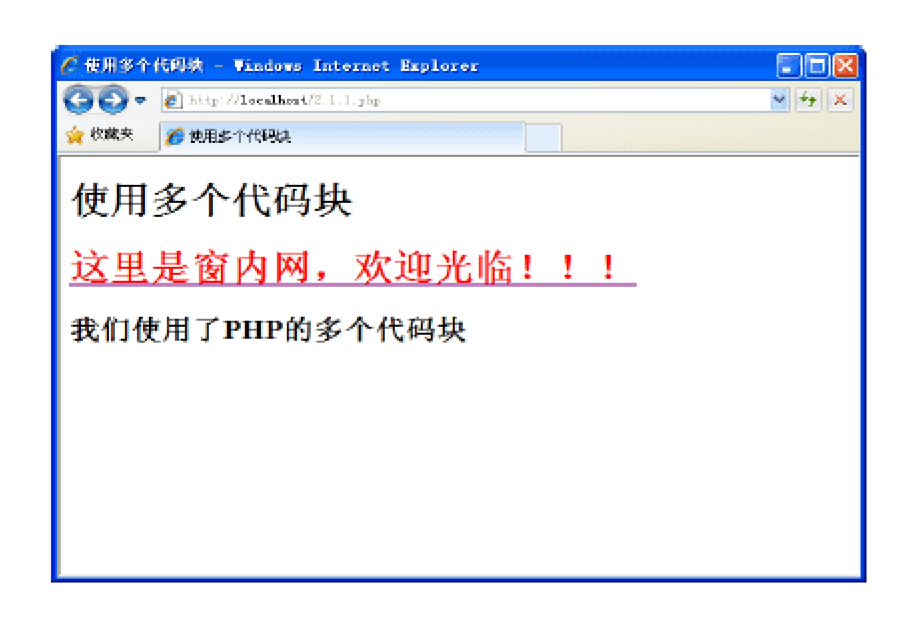


图 2-1 嵌入多个代码块效果

2.1.5 实例分析



源码解析：

上述实例中，我们使用多种 PHP 的界定标记在一个 PHP 页面中嵌入代码，在这几种标记中以默认标记的使用为主。在 PHP 代码中，定义了一个名为$str 的变量，变量声明后可以被以后的代码重复使用。

2.2 PHP 注释

对于初学者来说，有良好的编程习惯是程序员最重要的品质。在程序中添加注释不仅能够使其他人阅读方便，还为以后的代码调试提供了便捷。在这一节中，我们将具体地介绍 PHP 的注释。



视频教学：光盘\videos\02\注释.avi　　长度：3 分钟

在 PHP 中，注释起到了功能说明的作用，有注释的程序更容易看明白。注释语句是一种非执行语句，程序在运行时不会把注释内容显示出来。我们可以使用多种风格的注释，例如单行注释、多行注释和 Perl 语言风格的注释等，如表 2-1 所示。

表 2-1　PHP 的注释形式

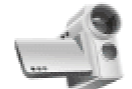
注　释	注释名称	注释描述
//	单行 C++语法	如果加入的注释没有超过一行，这时我们可以选择使用单行注释。因为这种注释很短，没有必要界定这种注释的结束。被注释的内容从//开始，到当前行结束
#	shell 语法 (Perl 风格)	单行注释形式。被注释的内容从#开始，到当前行结束
/* … */	多行 C 语法	被注释的内容从/*开始，到*/结束适用于注释内容包括多行的情况，像详细的功能描述

在实际使用过程中，这几种注释可以根据需要同时出现在一个程序中。例如：

```
<?php
    /* 这是个注释的例子
    时间：2010-12-19 16:40
    作者：王亚斌
    */
    echo "第一种注释.\n";// 单行注释，C++语法的注释
    echo "第二种注释.\n";# Perl 语言风格的注释
?>
```

2.3　输出 PHP 嵌入标记<?php?>

使用 PHP 技术可以使程序在客户端和服务器端具有高度的交互性，从服务器得到的信息要输出到页面上，通常使用 PHP 内置的输出函数。



视频教学：光盘\videos\02\输出函数介绍.avi　　长度：7 分钟

2.3.1　基础知识——输出函数

在 PHP 中，把信息输出到客户端通常采用输出函数。PHP 有自带的内置函数，这些函数的功能很方便，也很强大。主要有 print()、echo()、printf()和 sprintf()函数。

1. print()函数

print()函数的主要作用是向页面输出信息，为用户提供信息反馈。该函数能够输出字符串和变量，它的返回值是一个布尔类型，如果成功输出，则返回 true。其语法格式如下所示：

```
boolean print(argument)
```

根据上面的语法格式，我们创建一个示例，来具体地讲解这个函数的使用，代码如下：

```
<h1>
```

```
<?php
print("PHP 是一个不错的语言<br>");?>
<?php
  $a="PHP";
  print($a."是一个不错的语言");
?>
</h1>
```

在以上代码中，创建了一个变量$a，然后使用“.”将字符串变量连接(圆点代表字符串连接符)。在服务器中执行这个文件，我们可以看到如下结果。

```
PHP 是一个不错的语言
PHP 是一个不错的语言
```

2. echo()函数

echo()函数的作用和 print()函数是相同的，都是向客户端输出信息。但不同的是，print()函数返回一个 boolean 值，可以用于复杂的表达式中，而 echo()函数返回的是 void 值，不能应用于复杂表达式，但可以输出多个字符串，执行速度上要比 print()函数快一些。该函数的语法格式如下所示：

```
void echo(string $args[,string $args2 …]);
```

我们可以使用这个函数向页面输出比较简单的字符串，如以下代码所示：

```
<?php
     echo "在这里使用 echo 函数输出";
     echo "在这里使用 echo 函数输出";
  ?>
```

3. printf()函数

printf()函数的功能和 print()函数相同，也是向客户端输出字符串，但该函数向客户端输出的字符串是经过格式化的。其语法格式如下所示：

```
boolean printf(string $format[,mixed args])
```

args 是指定的输出参数，但它的输出是经过 format 格式化的。format 参数可以对输出数据进行合理而充分的控制，如精度、对齐方式、位置和类型等。这个参数是由 5 部分组成的，如表 2-2 所示。

表 2-2　format 参数组成

名　称	位　置	详细信息
填充提示符	第 1 个	这一部分确定为达到正确的字符串大小所用的填充符。默认为空格，也可以指定其他填充提示符，用法可选
对齐提示符	第 2 个	这一部分确定输出是左对齐还是右对齐。默认为右对齐，可以用一个符号表示左对齐，用法可选

续表

名　称	位　置	详细信息
宽度提示符	第 3 个	这一部分确定此函数输出的最小字符数，用法可选
精度提示符	第 4 个	这一部分确定显示的小数位数，但只对浮点数类型的数据有效，用法可选
类型提示符	第 5 个	这一部分确定如何转换参数

这些参数所支持的类型提示符有很多，如表 2-3 所示，我们将进行详细的讲解。

表 2-3　支持的类型提示符

提示符类型	详细信息
%b	将参数认定为一个整数，显示为二进制数
%c	将参数认定为一个整数，显示为对应的 ASCII 字符
%d	将参数认定为一个整数，显示为有符号的十进制数
%f	将参数认定为一个浮点数，显示为浮点数
%o	将参数认定为一个整数，显示为八进制数
%s	将参数认定为一个字符串，显示为字符串
%u	将参数认定为一个整数，显示为无符号的十进制数
%x	将参数认定为一个整数，显示为小写的十六进制数
%X	将参数认定为一个整数，显示为大写的十六进制数

接下来我们看看以下这几个小例子，详细地了解一下该函数的用法，代码如下所示：

```
<?php
    printf("%.4f",3.12);
    printf("I like %s","php<br>");
    printf("%1\$s like %2\$s ,he like %2\$s","I","PHP");
  ?>
```

4. sprintf()函数

sprintf()函数和 printf()函数一样，都是输出格式信息，由于该函数与 print()函数的功能基本相同，语法格式如下所示：

```
string sprintf(string $format[,mixed args])
```

下面我们通过一个小示例了解该函数的用法，代码如下：

```
<?php
   $a=sprintf("$%01.2f",56.8);
    echo $a;
  ?>
```

2.3.2 实例描述

我们在做一些小型的 PHP 技术论坛时，公告板是必不可少的一个模块，但是我们要在公告板中输出像<?php ?>这样的数据时，如果不经过处理的话，程序会把它当成 PHP 中的界定符而被编译器忽略，在公告板中就看不到想要输出的效果。下面就让我们通过一个实例来详细了解输出的使用。

2.3.3 实例应用

【例 2-2】 输出 PHP 嵌入标记<?php ?>

(1) 首先，新建一个名为 out.php 的文件。

(2) 在 out.php 文件中，要输出 php 嵌入标记的地方添加如下代码：

```
<h2>
使用 PHP 的嵌入标记如
<?php
echo htmlentities( '<?php' );
echo htmlentities( '?>' );
?>
可以在页面中嵌入 php 代码.
</h2>
```

(3) 保存好要修改的代码。

2.3.4 运行结果

打开服务器，在浏览器中执行文件，可以看到公告板中已经可以输出我们想要的 PHP 嵌入标记了，如下所示。

```
输出结果:
使用 PHP 的嵌入标记如 <?php?>可以在页面中嵌入 php 代码.
```

2.3.5 实例分析



源码解析：

上述实例中，我们使用正常的输出函数 echo()输出 PHP 默认的嵌入标记常常被编译器忽略掉，因此要使用 PHP 的 htmlentities()函数把字符转换为 HTML 实体，这样就能输出嵌入标记了。

2.4 数 据 类 型

数据类型是指具有相同特性的一类数据的统称。在 PHP 中比较常见的数据类型有整型、

字符串型、浮点型和布尔型，PHP 5 中还会有更多的补充。

 视频教学：光盘\videos\02\数据类型.avi　　　　长度：21 分钟

2.4.1　基础知识——数据类型

本节主要介绍 PHP 中的标量数据类型、类型之间的转换方法，以及可用于类型的函数。

1. 标量数据类型

在 PHP 中，标量数据类型只能包含单一数据信息，如包含了整型数据，就不能包含字符串信息。常见的标量数据类型为：布尔型、浮点型、整型和字符串型，如表 2-4 所示。

表 2-4　标量数据类型

名　称	描　述	示　例
布尔型	boolean 值可以为 true 或 false，这两个值不区分大小写。也可以用 0 表示 false，用非 0 值表示 true	$a = false; $b = true; $c = 0; $d = 1;

整型	不含小数位的数，最多为 32 位数。integer 数是集合 Z = {..., −2, −1, 0, 1, 2, ...} 中的一个数。整型数可以用十进制、十六进制或八进制符号指定，并且前面可以加上“+”号或者“-”号。如果用八进制符号，数字前必须加上 0(零)，用十六进制符号数字前必须加上 0x	$a = 123; // 十进制数
		$a = −123; // 十进制负数
		$a = 0123; // 八进制数
		$a = 0x1A; // 十六进制数

浮点型	浮点数也称为单精度数(float)、双精度数(double)或实数(real number)，可以指定包含小数部分的数	$a = 1.234; $a = 1.2e3; $a = 7E−10;
字符串	字符串是一系列字符的组合，可以使用 3 种方式来定义，分别为：单引号、双引号、定界符	$a = "字符串";

在使用字符串时，有时需要使用反斜线(\)进行转义才能使字符串正确地输出。例如使用单引号创建的字符串，就需要进行转义。在 PHP 中我们常用到的转义字符如表 2-5 所示。

表 2-5　转义字符

转义字符序列	详细信息
\n	表示换行
\r	表示回车
\t	表示水平制表符
\\	表示反斜线
\$	表示美元符号
\”	表示双引号
\[0-7]{1,3}	此正则表达式序列用于匹配一个八进制符号表示的字符
\x[0-9A-Fa-f]{1,2}	此正则表达式序列用于匹配一个十六进制符号表示的字符

例如以下例子：

```
<?php
        echo '字符串';                    // 输出：字符串
        echo ' "I\'m jack"';              // 输出：这个字符串："I'm jack"
        echo '请修改并保存 C:\\*.*?文件'; // 输出：请修改并保存 C:\*.*?文件
    $a = "字符串";
    echo $a;                              // 输出：字符串
?>
```



如果在单引号之前或字符串结尾需要出现一个反斜线，需要用两个反斜线表示。注意如果试图转义任何其他字符，反斜线本身也会被显示出来。所以通常不需要转义反斜线本身。

2. 数据类型转换

数据类型转换就将一种数据类型转换成另一种数据类型。这种数据类型之间的转换分两种：一种是强制转换；一种是自动转换。在本小节中我们将详细地介绍这两种类型的转换。

1) 强制数据类型转换

强制数据类型转换是指将一个变量强制转换为与原类型不相同的另一种类型的变量。这种转换的实现非常简单，只需在变量前加上要转换的数据类型即可，转换过后就可以按照转换过的类型来计算。其转换操作符如表 2-6 所示。

表 2-6　强制转换操作符

转换操作符	转换类型
(array)	数组
(bool)或者(Boolean)	布尔值
(int)或者(integer)	整数
(object)	对象
(real)或者(double)或者(float)	浮点数
(string)	字符串

上面介绍了强制转换操作符，它既可以将一个整数转换为双精度的数字，也可以转换一个对象。下面来看一个示例，代码如下：

```
<?php
 $b = 12.2;
 $c =(int)$b;
 echo $c."<br>";
 $a = 5;
 $b = (double)$a+4;
 echo $b."<br>";
 $str = "我喜欢 PHP</a>";
 $i =(int) $str;
 print $i."<br>";
 $ob = (object) $str;
```

```
print $ob->scalar;
?>
```

在代码中，我们可以看到，既可以将整数转换为浮点型，也可以将一个浮点型转换为整型。如果将双精度的浮点型转换为整型，双精度数的小数部分会被截断，直接向下取整。当将一个字符串转换为整型时，它返回的值是 0，没有多大的意义。而将整数转换为字符串时，将不做任何修改直接输出。在 PHP 中，任何类型的数据都能转换为对象，转换的结果是该变量成为这个对象的属性，属性名为 scalar，使用到这个变量时，直接使用对象调用属性就可以了，接下来执行代码，我们可以看到如下结果：

```
12
9
0
我喜欢 PHP
```

2) 自动数据类型转换

在 PHP 中，数据类型可以根据变量所处的环境来自动转换为适应的类型，由于 PHP 语言是一种弱类型的语言，对类型的定义比较松散，所以数据类型的自动转换将非常地实用。

接下来我们看一个小示例，代码如下：

```
<?php
/*
 * Created on 2010-12-21
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
 $aa = 65;
 $bb = "128yy";
 $sum = $aa+$bb;
 echo $sum;
 echo "<br>";
 $g = "PHP 的数据类型";
 if($g){
    echo $g."被转换为了布尔值";
    }
?>
```

在本示例中，整型变量和带有数字的字符串变量相加时，其结果值会自动转换为整型变量，如代码中的$aa 和$bb 相加，结果就是一个整型变量$sum。而在条件语句中，字符串变量$g 就转换成布尔类型的变量。执行代码后，我们可以看到如下结果。

```
193
PHP 的数据类型被转换为了布尔值
```

3. 类型函数

在 PHP 中，类型函数是由与类型相关的函数和类型标识符函数组成的，我们将在这个小节里详细地介绍这些函数。

1) 与类型相关的函数

我们都知道，PHP 语言是一种弱类型的语言，在进行变量声明时，不需要带任何的数据类型。在实际操作过程中，需要获取某个变量的数据类型或设置某个变量的数据类型时，我们需要用到 PHP 的内置函数。PHP 中有两个获取数据类型的函数，分别是 settype()函数和 gettype()函数。下面我们将对这两个函数进行详细的介绍。

(1) settype()函数。

settype()函数的功能是将变量设置为指定的数据类型，其语法格式如下：

```
boolean settype(maxed $var,string $type)
```

从该函数的语法格式中，我们可以知道函数如果执行成功将返回真，失败时将返回假。来看下面这个例子，其代码如下：

```
<?php
/*
 * Created on 2010-12-21
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
 $str = "1987 年";
 $fal = true;

 settype($str,"integer");
 echo $str;
 echo "<br>";
 settype($fal,"string");
 echo $fal;
?>
```

在本例中，声明了一个字符串类型的变量$str 和一个布尔类型的变量$fal，通过函数 settype()设置变量$str 的类型为整型，设置变量$fal 的类型为字符串型，分别输出这两个变量，我们可以在页面中看到如下所示的结果：

```
1987
1
```

(2) gettype()函数。

gettype()函数主要用于获取指定变量的数据类型，并且返回 string 类型的数据，返回值为变量的数据类型。该函数的语法结构如下所示：

```
string gettype(mixed $var)
```

该函数只有一个指定类型的参数，这个参数指定了要获取数据类型的变量。我们来看一个例子，代码如下：

```
<h1>
<?php
/*
```

```
 * Created on 2010-12-21
 *
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
 $lastname = "王亚斌";
 $id = 12427;
 echo "ID 号: ".$id;
 echo "<br>";
 echo "类型是: ".gettype($id)."<br>";
 echo "姓名: ".$lastname;
 echo "<br>";
 echo "类型是: ".gettype($lastname);
?>
</h1>
```

执行上述代码，我们可以看到如图 2-2 所示的结果，$id 变量的数据类型是 integer 型，而 $lastname 变量的数据类型是 string 类型的。

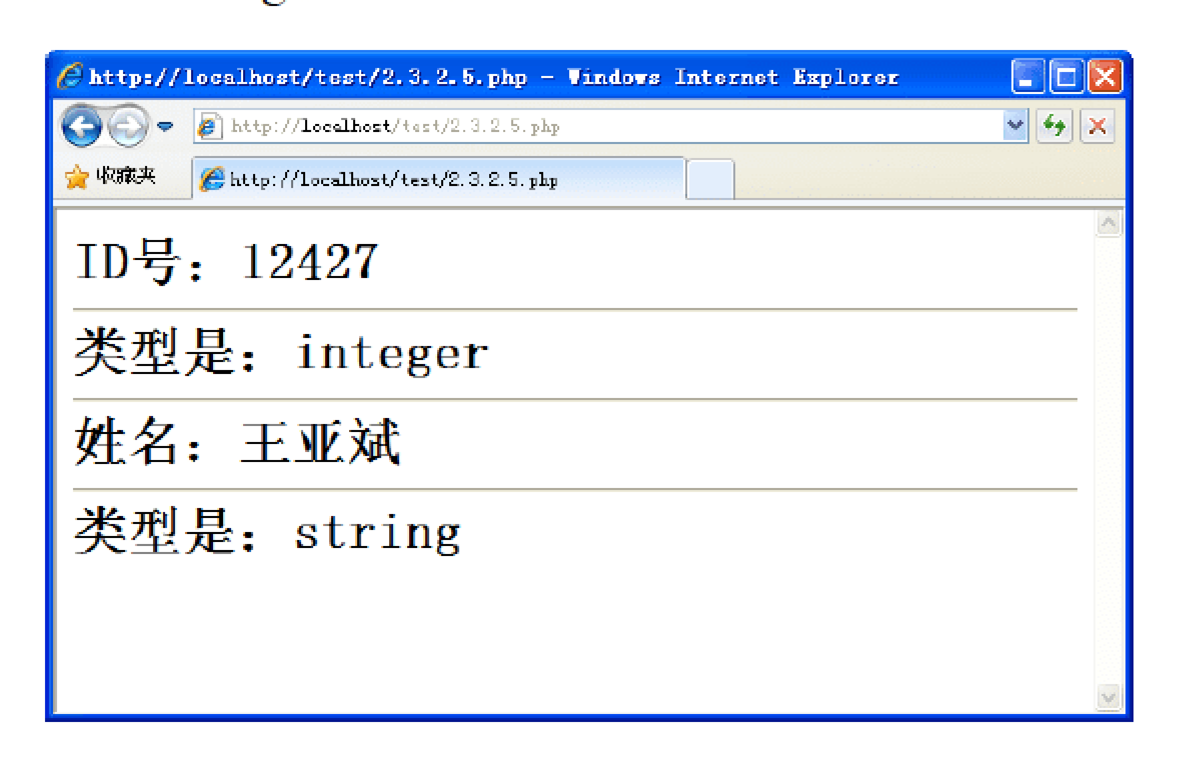


图 2-2　gettype()函数运行结果

2) 类型标识符函数

类型标识符函数是用来判断变量数据类型的函数，这种函数有许多，如 is_bool()、is_array()、is_float()、is_null()、is_number()、is_object()、is_resource()、is_scalar()和 is_string()等。这些函数都有相同的参数、返回值和命名约定，所以我们以 is_array()函数为例讲解这些函数的具体使用方法。

is_array()函数用来确定变量的类型是否为数组，其语法结构如下：

```
boolean is_array(mixed $var)
```

如果确定变量的数据类型是数组，则返回值为 true，否则其返回值为 false。下面我们来看一个小例子，代码如下：

```
<h1>
<?php
 $arr = array("足球","篮球","乒乓球","羽毛球","排球");

 if(is_array($arr)){
```

```
    echo "变量 arr 是一个数组类型<br>";
    print_r($arr);
 }
?>
</h1>
```

执行上述代码，我们可以看到如图 2-3 所示的结果。

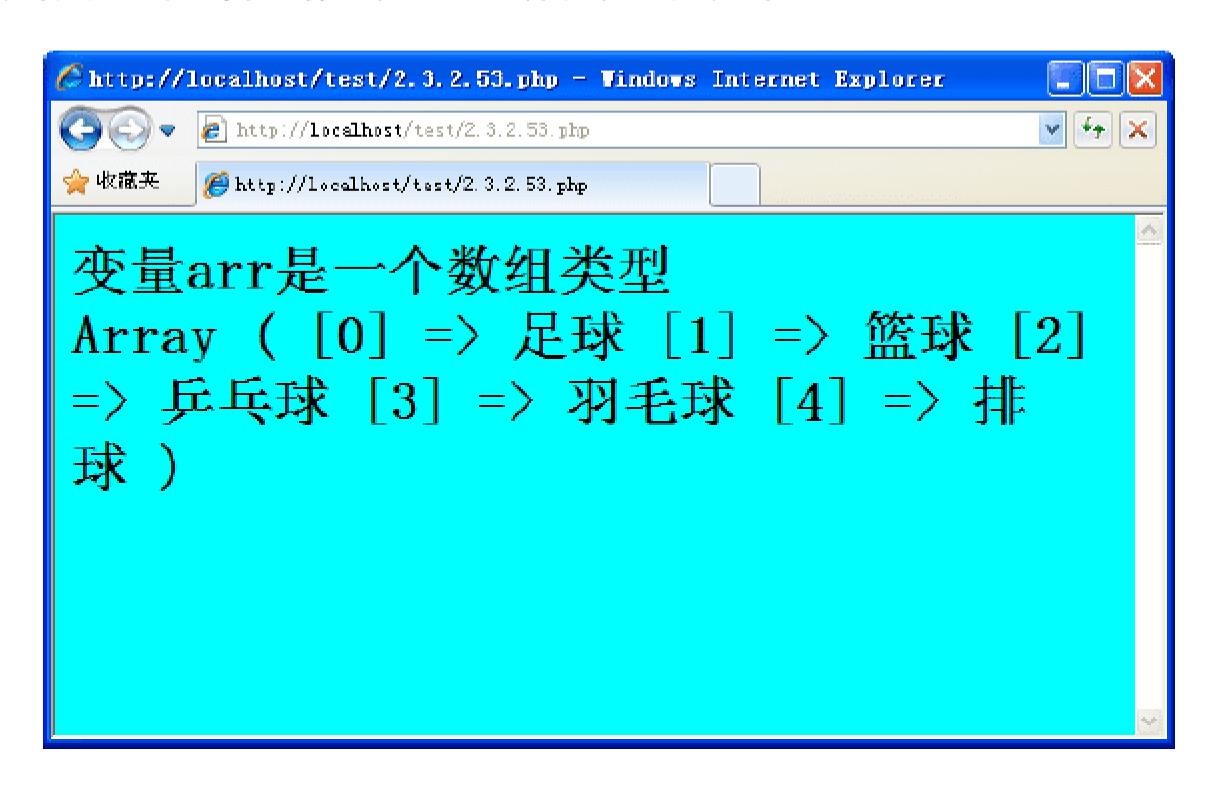


图 2-3 类型标识符函数的运行结果

2.4.2 实例描述

在实际应用中，PHP 的程序要对页面的数据进行处理，例如登录页面中的用户名和密码等数据信息，我们在使用 PHP 对这些数据进行处理时，也是要注意它们的数据类型的。下面通过一个登录页面来讲解一下数据的使用。

2.4.3 实例应用

【例 2-3】 管理员登录

(1) 首先新建一个名为 login.php 的页面，在这个页面中，我们在 form 表单中需要用到两个文本输入框，分别用来输入用户名和密码；还需要用到两个按钮，分别是【提交】按钮和【重置】按钮。表单中的数据以 post 方式进行提交，提交到目标页面 post.php 后进行处理数据。其代码如下：

```
<body bgcolor=#008099>

<div id="bigbox"  class="ceng">
<div id="main" class="ceng">
<form id="form1" name="form1"  method="post" action="post.php">
<div id="log">
<div id="yonghu">用户名:
  <input name="user" type="text" id="user" size=25 />
</div>
<div id="mima">密  码:
  <input name="pass" type="password" id="pass" size=27/>
```

```
</div>
</div>
<div id="tijiao" >
  <input type="submit" name="Submit" value="提交" />
  <input type="reset" name="Submit2" value="重置" />
</div>
</div>
</form>
</div>
</div>
</body>
```

(2)　定义 login.php 的页面样式，其代码如下：

```
<style type="text/css">
<!--
body,td,th {
 font-size: 14px;
}
a:link {
 text-decoration: none;
 color: #FFFFFF;
}
a:visited {
 text-decoration: none;
 color: #000000;
}
a:hover {
 text-decoration: none;
}
a:active {
 text-decoration: none;
}
.STYLE99 {

 font-family: Georgia, "Times New Roman", Times, serif;
 color: #FF00FF;
}
-->
</style></head>
```

(3)　新建页面 post.php，由于还没有讲到数据库，在这里暂不使用与数据库的链接。我们只在页面中定义两个变量，$name 和$password。代码如下所示：

```
<title>登录信息</title>
<body bgcolor=#008099>
<?php
/*
 * Created on 2010-12-21
 *
```

```
 * To change the template for this generated file go to
 * Window - Preferences - PHPeclipse - PHP - Code Templates
 */
$name = "wangyabin";
$password = "123456789";
if($_POST[user]==$name && $_POST[pass]==$password){
   echo "登录成功";
}else{
     echo "对不起,您登录失败! 请重新登录.";
}
?>
</body>
```

2.4.4　运行结果

在 Apache 服务器开启的前提下，先运行 login.php 页面，我们可以看到如图 2-4 所示的效果。

当填写好用户名和密码后，单击【提交】按钮，根据 form 中的 action 路径，前台传递的数据会到 post.php 中进行处理，如果用户名和密码正确，则跳转到如图 2-5 所示的页面。

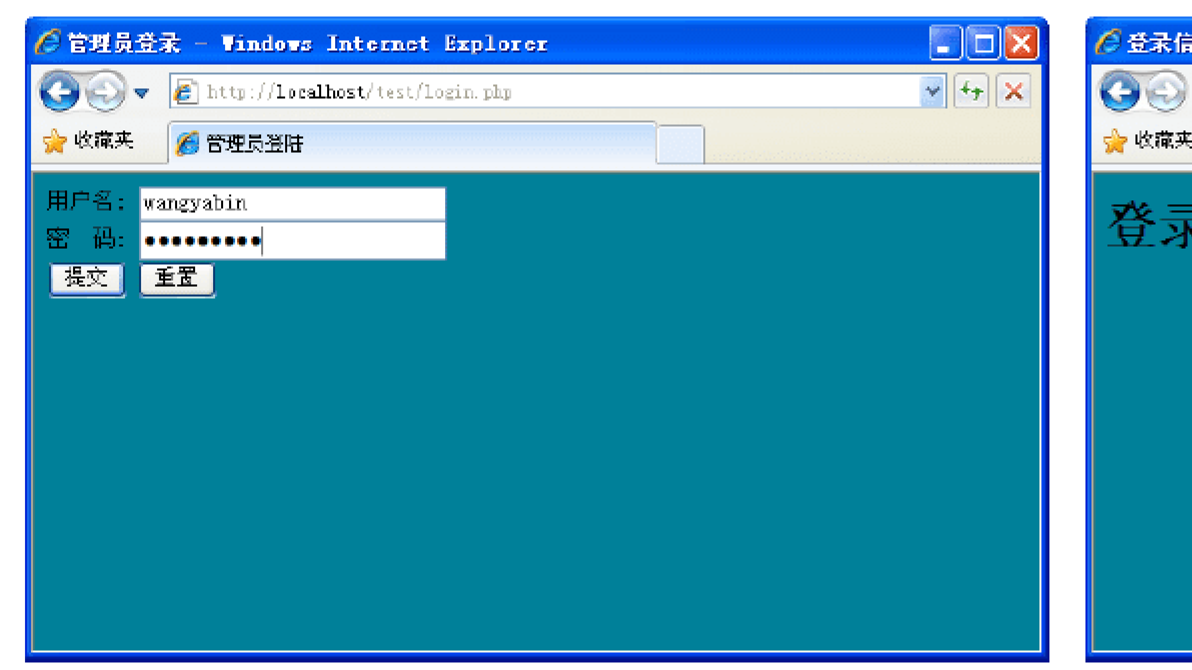


图 2-4　管理员登录

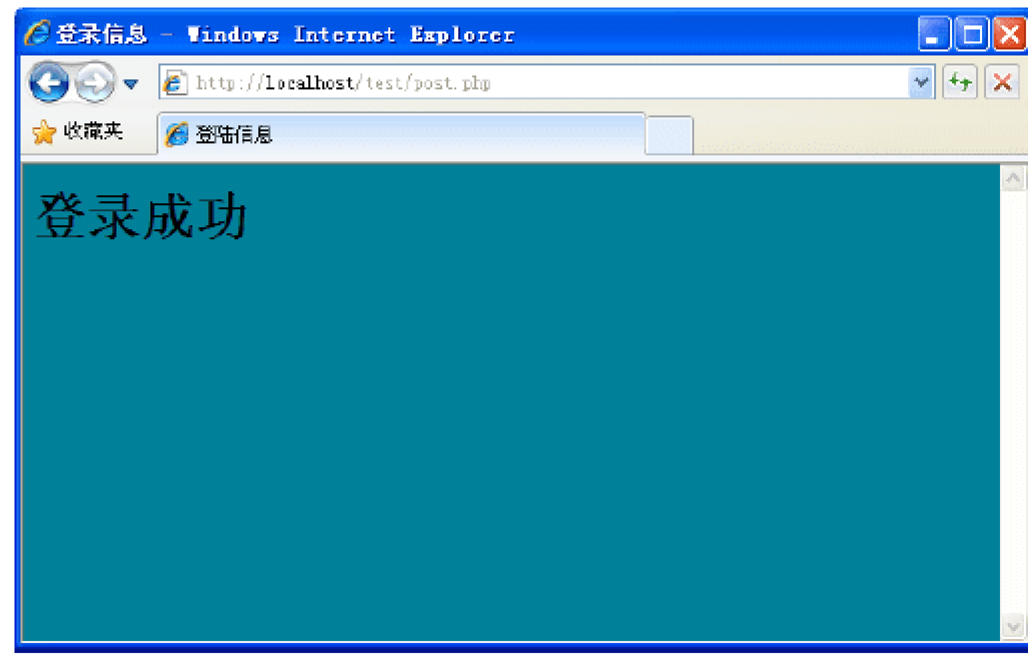


图 2-5　登录成录

2.4.5　实例分析



源码解析：

上述实例中，我们在 post.php 中建立了字符串类型的数据，在 if 语句中，我们把前台页面 login.php 传递的 user 和 pass 这两个数据进行对比，如果相同则返回“登录成功”。

2.5　接收表单变量

在 PHP 程序运行过程中随时可以变化的量叫做变量。在变量中既可以存放字符串、数值和日期，也可以存放属性。这里需要强调的是变量的值是临时的，当系统运行时，变量的值是存在的，当程序结束后，变量的值就会被当做垃圾在内存中回收。本节将详细讲解变量的创建、

调用和变量的作用域。

 **视频教学：**光盘\videos\02\变量.avi　　　　长度：7 分钟

2.5.1　基础知识——变量

PHP 中的变量对于初学者来说，有些难以理解，因为在多种语言中，变量的声明都要声明其类型，而 PHP 的变量则是不需要指定类型的。

1．变量的创建和调用

在 PHP 中，我们需要通过美元符号$来作为前缀，$之后的为变量的标识符，标识符是一个区分不同对象的符号，如变量的名称、函数的名称等。

PHP 是一种弱类型语言，在声明变量时，不需要显示声明变量的类型，变量可以是任何类型的值。在创建变量时，首先要声明变量，然后再给变量赋值。这个值可以是任何类型的，变量只有在程序运行时才对其类型进行检查，并且可以用一个不同类型的值来取代原有的值，例如以下代码：

```
<?php
 $name = "张小明";
 $name = 123;
?>
```

变量的赋值方式有两种：一种是值赋值；另一种是引用赋值。值赋值就是直接通过赋值表达式将一个值赋给变量，通过这种方式可以把原来的值覆盖掉。引用赋值是指所创建的变量与另一个变量引用的内容相同。当需要使用到引用赋值时，只需在等号后面加一个&符号即可，例如如下代码：

```
<?php
$name = "王亚斌";
 $no = "41272419874546 51";
 echo "姓名：".$name."<br>";
 echo "<hr>";
 echo "学号：".$no."<br>";
 echo "<hr>";
 $a1 = "1";
 $a2 =& $a1;
 $a2 = "2";
 echo "年龄：";
 echo $a1;
 echo $a2;
?>
```

在代码中我们可以看到，$name 和$no 采用了值赋值的形式为这两个变量进行赋值。而$a2 采用了引用赋值，当两个变量之间采用了引用赋值后，这两个变量就成为一体，只要一个变量的值改变，另一个变量的值也会随之改变。

2. 变量的作用域

变量的作用域是指变量在程序中的作用范围，变量在声明时的位置决定了变量的作用域。在 PHP 中，虽然可以在脚本的任何位置声明变量，但变量的作用域大致可以分为 4 类，分别是局部变量、函数参数、全局变量和静态变量。下面详细讲解这几种变量的作用域。

1) 局部变量

在函数中声明的变量叫做局部变量，声明后只能在函数内部使用，不能在函数外部被访问，如果在函数外部赋值，将被视为另外一个变量。例如以下代码：

```
<?php
 $pay = 10000;
  function pay(){
     $pay1 = 10000;
     $pay2 = 5000;
     $pay = $pay1+$pay2;
     echo "函数内的工资是：".$pay;
     echo "<br>";
     echo "<hr>";
  }
  pay();
  echo "函数外的工资是：".$pay;
?>
```

在这段代码的 pay()函数中声明了一个局部变量，在函数外部也声明了一个同名的变量，由于这两个变量的作用域不同，所以输出的结果也不同，如下所示：

```
函数内的工资是：15000
函数外的工资是：10000
```

2) 函数参数

PHP 的函数也可以和其他编程语言一样，通过参数的形式来介绍外部的值。但这些参数必须在函数的首部进行声明，当推出函数时，这些参数将无法访问，而参数的值也会消失。我们来看以下代码：

```
<?php
  function info($name,$sex,$age){
     echo "您的姓名是：".$name;
     echo "<br>";
     echo "您的性别是：".$sex;
     echo "<br>";
     echo "您的年龄是：".$age;
     echo "<br>";
  }
  info("王亚斌","男",24);
?>
```

函数参数也可以当做局部变量，但当函数撤销时，参数也会被撤销。运行上述代码，其结果如下所示：

```
您的姓名是：王亚斌
您的性别是：男
您的年龄是：24
```

3)　全局变量

全局变量是指在任何地方都可以访问的变量，但是要创建一个全局变量，不许进行显示的声明方法很简单，只需在要声明的变量前加上 PHP 关键字 global 即可。用法如以下代码所示：

```
<?php
  $num = 10;
  function info(){
    global $num;
    $num+=5;
    echo "计算之后的数值是：".$num;
  }
  info();
?>
```

代码运行后的结果为：

```
计算之后的数值是：15
```



global 是声明全局变量的，声明后的变量一旦在一个地方修改了值，其他地方的值也会随之改变。

4)　静态变量

PHP 中还有一种变量，一旦声明并赋值后，其值不会再改变也不会丢失，这种变量叫做静态变量。在声明变量时只需在其前面加上关键字 static 即可，如以下代码所示：

```
<?php
  function info(){
     static $num = 10;
    $num++;
    echo "计算之后的数值是：".$num;
    echo "<br>";
  }
  info();
  info();
  info();
?>
```

上述代码的运行结果如下：

```
计算之后的数值是：11
计算之后的数值是：12
计算之后的数值是：13
```

我们从结果中可以看到，当静态变量声明后，将会被保存到内存中，每次执行函数时都保留前面的值。

2.5.2　实例描述

PHP 中变量的使用是相当普遍的，例如在处理表单中的数据时，PHP 就会获取表单传递的变量值。有一天，一个朋友问我，在 PHP 中接收表单传递过来的变量值有几种方式，他在页面之间传递变量值的问题总是很迷惑。正好今天做了一个用 PHP 处理表单的例子，在这里就向大家展示一下吧，也希望能给我的那位朋友解惑。

2.5.3　实例应用

【例 2-4】 接收表单变量

(1) 首先，新建一个 registerl.html 页面。在页面中添加一个 form 表单和 4 个输入框，分别用来输入用户名、密码、密码确认和邮箱，还有一个【注册】按钮。然后将表单以 POST 方式提交到 register.php，部分代码如下：

```
<form action="./register.php" method="post" id="apForm">
    <input type="hidden" name="referer" value="./index.html" />
    <div class="content">
      <p>
        <label for="email"> 您的登录名：</label>
        <input class="input" type="text" id=username maxLength=32 onBlur=
"checkusername();" name="username">
        <span id="checkusername"> </span> </p>
      <p>
        <label for="email"> 登录密码：</label>
        <input  class="input"  style="width:148px;" onkeyup=EvalPwdStrength
(this.value); onBlur="checkpassword()" type=password maxLength=16 id="password"
name="userpass">
        <span id="checkpassword"> </span> </p>

      <p>
        <label for="email"> 密码确认：</label>
        <input class="input" style="width:148px;"  onBlur="checkpassword2()"
type=password maxLength=16 id="password2" name="userpass2">
        <span id="checkpassword2"> </span> </p>
      <p>
        <label for="email"> 电子邮箱：</label>
        <input id=email maxLength=32  onBlur="checkemail()" class="input"
name=email>
        <span id="checkemail"> </span> </p>


      <p>
        <input type="hidden" name="registersubmit" value="true">
        <input type="submit" name="mysubmit" value="" class="tj">
      </p>
```

```
    </div>
  </form>
```

(2) 新建一个名为 Register.php 的页面，在此页面中，我们使用 PHP 中的两种页面之间转递过来的变量值的方法来接收表单中的数据，并进行简单的验证，其代码如下所示：

```
$name=$_POST[username];
 $pass=$_POST[userpass];
 $pass2=$_POST[userpass2];
 $email=$_POST[email];
 echo $username;
 if($name=="wangyabin"){
    echo "该用户名已经被注册了";
 }else{
    echo "您注册的用户名有效";
 }
```

2.5.4　运行结果

在服务器中运行 Registerl.html，我们可以看到如图 2-6 所示的注册页面。



图 2-6　注册页面

在注册页面中，输入用户名、密码和邮箱，然后单击【注册】按钮，表单中的值就会传到 Register.php 的页面进行接收和处理，处理结果如图 2-7 所示。



图 2-7　处理结果页面

2.5.5 实例分析



源码解析：

在本实例中，首先运行了注册页面 Registerl.html。当我们在表单中输入信息并单击【注册】按钮后，这些信息会以 POST 方式提交到 Register.php 页面。

在 Register.php 页面使用$_POST[varname]方式获取表单中的值，然后进行判断并输出结果。

2.6 使用 PHP 常量调试代码

在调试代码时，如果能够确切地知道我们的代码调试到哪一行，对程序员来说将有很大帮助。本节将通过学习 PHP 的常量来实现这个功能。



视频教学：光盘\videos\02\常量.avi　　长度：17 分钟

2.6.1 基础知识——常量

常量是指程序中无法修改的值，如 PI 等。常量需要用 define()函数来定义，常量一旦定义，就无法在程序中修改或重新定义，在程序中处理不需要修改的值时，我们会使用常量。声明常量的语法如下。

```
boolean define(string name,miced value)
```

我们来看下面这个例子。

```
<?php
  define("PI",3.1415926);
  echo "PI 的值为：".PI;
?>
```

上述代码中使用 define()函数定义了一个数学常量 PI，并赋值为 3.1415926，执行代码后可以看到以下结果。

```
PI 的值为：3.1415926
```

在 PHP 中还有很多的预定义常量，如常量__file__等，如表 2-7 所示。

表 2-7 PHP 预定义常量

常 量 名	说 明	注 意
__file__	当前 PHP 程序文件名	file 前后都是双下划线
__line__	当前执行语句在 PHP 程序文件中的行数	line 前后都是双下划线
PHP__VERSION	当前 PHP 的版本号	PHP 必须大写
PHP_OS	当前所用的操作系统类型	PHP 必须大写

续表

常 量 名	说　明	注　意
E_ERROR	指明最近一次产生不可恢复的错误	大写
E_WARING	指出有错误，但程序可以继续	大写
E_PARSE	语法错误，分析器将被停止分析	大写
E_NOTICE	产生异常，但不一定是错误，程序可以继续	大写

2.6.2　实例描述

有时我们在调试代码时，并不能确切地知道所编写的代码执行到哪一行，当程序出现错误时，只能了解到在整个文件中有错误，但不知道是哪一行出现的。这样就需要我们能确切地知道我们的代码执行到了哪一行，进而找到错误。在 PHP 中一些预定义的常量能够解决这一问题，下面我们来看一个实例。

2.6.3　实例应用

【例 2-5】 使用 PHP 常量调试代码

(1) 新建一个 error.php 文件，在里面写入 PHP 代码。

(2) 当我们的程序出现错误时，在代码中定义一个名为 report_error()的函数，然后执行该函数，就可以得到我们的程序当前执行到了哪一行。代码如下所示：

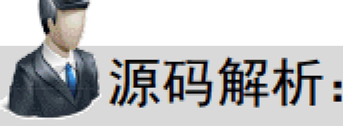
```
function report_error($file, $line, $message) {
echo "An error occured in $file on line $line: $message.";
}
report_error(__FILE__,__LINE__, "Something went wrong!");
```

2.6.4　运行结果

保存文件，在服务器运行时，执行文件，我们可以得到如下结果：

```
An error occured in D:\www\test\reg\error.php on line 18: Something went wrong!.
```

2.6.5　实例分析



源码解析：

在上述实例中，report_error()函数定义了三个参数。我们使用了 PHP 中的__FILE__和__LINE__常量，得到了正在执行文件的路径和当前代码执行的行数。

2.7 使用操作符验证表单数据

在 PHP 中，表达式是至关重要的，而表达式是由操作数和操作符组成的，其中最重要的当然是操作符的使用，因此，在进行 PHP 开发时熟练掌握操作符的使用方法是非常有必要的。

视频教学： 光盘\videos\02\算术操作符.avi　　长度：8 分钟
光盘\videos\02\赋值操作符.avi　　长度：5 分钟
光盘\videos\02\比较运算符.avi　　长度：3 分钟
光盘\videos\02\逻辑运算符.avi　　长度：5 分钟
光盘\videos\02\运算优先级.avi　　长度：7 分钟

2.7.1 基础知识——操作符

操作符是书写表达式时用于表示表达式中某个动作的符号，如两个数相乘的表达式等。在 PHP 中，常见的操作符有算术操作符、赋值操作符、逻辑操作符、比较操作符等。PHP 是一种和其他类型的语言不同的弱类型语言，因此操作符两侧的值会自动进行转换。下面我们来了解一下这些操作符。

1. 算术操作符

在 PHP 中，算数操作符主要用于数与数之间的和、差、积、商。常见的算数操作符如表 2-8 所示。

表 2-8　算数操作符

操 作 符	名　称	例　子	意　义
+	加法	$m+$n	取两数之和
-	减法	$m-$n	取两数之差
*	乘法	$m*$n	取两数的积
/	除法	$m/$n	取两数的商
%	取模	$m%$n	取两数相除后的余数
-	取负	-$n	取负数



在上述算术操作符中，除号(“/”)总是返回浮点数，即使两个运算数是整数(或由字符串转换成的整数)也是这样。取模运算$a % $b 在$a 为负值时，所得到的结果是负数。

2. 赋值操作符

赋值操作符的作用是将一个数据值赋值给一个变量。PHP 中常见赋值操作符如表 2-9 所示。

表 2-9　赋值操作符

操 作 符	名　称	例　子	意　义
=	等于	$m=8	$m 等于 8
+=	加等于	$m+=8	$m 等于$m 加 8
*=	乘等于	$m*=8	$m 等于$m 乘以 8
/=	除等于	$m/=8	$m 等于$m 除以 8
.=	并接等于	$m.=8	$m 等于$m 并接 8

3. 逻辑操作符

逻辑操作符用来判断某个对象是否符合标准，这种判断可能需要多个条件。因此逻辑操作符常常会被用在控制语句中。PHP 中常见的逻辑操作符如表 2-10 所示。

表 2-10　逻辑操作符

操 作 符	名　称	例　子	意　义
and	逻辑与	$m and $n	如果$m 与$n 都为 true，则结果为 true
or	逻辑或	$m or $n	如果$m 或$n 任意一个为 true，则结果为 true
!	逻辑非	！$m	如果$m 为不为 true，则结果为 true
&&	逻辑与	$m && $n	如果$m 与$n 都为 true，则结果为 true
||	逻辑或	$m || $n	如果$m 或$n 任意一个为 true，则结果为 true
xor	逻辑异或	$m xor $n	如果$m 或$n 任意一个为 true，但不能同时为 true，则结果为 true

逻辑操作符在控制语句中使用很方便，例如以下代码：

```
$mark = 90;
  if($mark<60){
    echo "成绩不及格";
  }
  if($mark>=60 and $mark<=90){
    echo "成绩中等";
  }
  if($mark>90 and $mark<100){
    echo "成绩优秀";
  }
```

在这段代码中，我们使用逻辑与操作符 and 来判断两条件是否同时为真，如果为真则输出相应的语句。

4. 比较操作符

比较操作符用来比较两个数值的大小，通常作为控制语句的判断条件，比较完成后，返回一个布尔类型的值。PHP 中常见的比较运算符如表 2-11 所示。

表 2-11　比较操作符

操 作 符	名　称	例　子	意　义
==	等于	$m==$n	如果$m 等于$n，则为真
===	全等于	$m===$n	如果$m 等于$n，并且它们的类型也相同，则为真
!=	不等	$m!=$n	如果$m 不等于$n，则为真
<>	不等	$m<>$n	如果$m 不等于$n，则为真
!==	非全等	$m!==$n	如果$m 不等于$n，或者它们的类型不相同，则为真
>	大于	$m>$n	如果$m 严格大于$n，则为真
<	小于	$m<$n	如果$m 严格小于$n，则为真
>=	大于等于	$m>=$n	如果$m 大于或者等于$n，则为真
<=	小于等于	$m<=$n	如果$m 小于或者等于$n，则为真

例如，在判断语句中使用比较操作符的代码如下：

```
$a = 10;
$b = 2;
if($a>$b){
   echo $a."大于".$b."<br>";
}
if($a<$b){
 echo $a."大于".$b;
}else{
 echo $a."是最大的";
}
```

5. 字符串操作符

字符串操作符经常用来对字符串进行操作。这种操作符有两个，一个是连接符“.”，另一个是连接赋值操作符“.=”，前者返回左右参数连接后的字符串，而后者则是将右边的参数附加到左边的参数之后，如表 2-12 所示。

表 2-12　字符串操作符

操 作 符	名　称	例　子	意　义
.	连接符	$m="我爱"."PHP";	$m 赋值为字符串“我爱 PHP”
.=	连接赋值操作符	$m.="王亚斌";	$m 等于它的当前值和“王亚斌”的并接结果

6. 位操作符

位操作符允许精细操作整数数据的每一个位，常见的位操作符如表 2-13 所示。

表 2-13　位操作符

操 作 符	名　称	例　子	意　义
&	按位与 and	$m&$n	将$m 和$n 中都为 1 的位设为 1

续表

操作符	名　称	例　子	意　义
|	按位或 or	$m\|$n	将$m 或$n 中为 1 的位设为 1
^	按位异或 Xor	$m^$n	将$m 和$n 中不为 1 的位设为 1
~	按位非 not	~$m	将$m 中为 0 的位设为 1，为 1 的位设为 0
>>	右移 shift right	$m>>$n	将$m 中的位向左移动$n 位，每次移动一次表示乘以 2
<<	左移 shift left	$m<<$n	将$m 中的位向右移动$n 位，每次移动一次表示除以 2



其中“~”是单目运算符，其余的都是双目运算符。对十进制数进行位运算时，要先转换为二进制数，然后再按规则进行运算。

7. 错误控制操作符

“@”是 PHP 支持的一个进行错误控制的操作符。在表达式的前面加上这个操作符，如果表达式出现错误，将不会输出错误信息。

8. 自增自减操作符

PHP 中的自增、自减操作符“++”、“--”的用法和 C++中的用法基本相同，唯一不同的就是 PHP 中的自增、自减操作符除了可以运用到数值型以外，还可以运用于字符串。另外“++”和“--”放到变量前和变量后，会产生不同的效果，如表 2-14 所示。

表 2-14　自增自减操作符

例　子	名　称	意　义
$m++	后自增	先返回$m，然后将$m 的值加 1
$m--	后自减	先返回$m，然后将$m 的值减 1
++$m	前自增	先将$m 的值加 1，再返回$m
--$m	前自减	先将$m 的值减 1，再返回$m

2.7.2　实例描述

很多玩游戏的人都知道，为了未成年人的身心健康，需要合理控制游戏时间。现在的游戏都会有防沉迷系统，这种系统的前提就是能够判定你是否已满 18 周岁，对于未满 18 周岁的未成年人，我们要采取一定的措施不能让其注册。在这里我们就会用到比较操作符。下面让我们来看一个实例。

2.7.3　实例应用

【例 2-6】 使用操作符验证表单数据

(1) 在第 2.5 节的 index.html 页面中，我们添加了以下代码，新建一个能够输入年龄的表单输入框，代码如下：

```
<p>
        <label for="age"> 您的年龄：</label>
        <INPUT class="input" type="text" id=age maxLength=32 name="age">
</p>
```

(2) 在 Register.php 中添加一些代码，如下所示：

```
if($age<=18){
    echo "您今年".$age.",对不起，您还未成年,这种游戏不适合您！";
}else{
    echo "恭喜您,游戏可以注册,但为了您的健康,请不要过度地玩游戏！";
}
```

(3) 保存页面。

2.7.4 运行结果

开启服务器，运行页面，可看到如图 2-8 所示的效果，在表单中输入信息。

在注册页面中，输入用户名、密码、邮箱和年龄，然后单击【注册】按钮，表单中的值就会传到 Register.php 的页面进行接收和处理，处理结果如图 2-9 所示。



图 2-8 注册页面



图 2-9 处理结果页面

2.7.5 实例分析



源码解析：

上述实例，在业务处理页面中我们主要使用了比较运算符“<=”来对前台页面提交过来的年龄数据进行比较，从而对程序的执行流程进行判断。

2.8 特殊数据类型

PHP 还有一些特殊的数据类型，这些数据类型主要提供某种特殊的用途，无法归入任何

类别。

 **视频教学：**光盘\videos\02\资源和空.avi　　　长度：25 分钟

PHP 的特殊数据类型主要包括资源和空(null)数据类型。

1. 资源

在 PHP 中，PHP 程序通常与一些外部数据源进行交互，如文件、数据库和网络流等。通常这种交互是通过句柄来完成的，这些句柄保持对资源的引用，直到通信结束，句柄才能撤销。这些句柄就属于资源数据类型。

并不是所有的函数都返回资源，只有 PHP 脚本中负责将资源绑定到变量的函数才能返回资源，而该资源类的变量也并不能真正地保存一个值，实际上只保存了一个指针，指向所打开的资源类型变量。在使用时，直接调用指针就行，如果输出其内容，我们将看到一个资源的 ID 号的引用。

2. 空(null)

空(null)就是“什么也没有”，没有值。它既不表示零，也不表示空格。这种定义理解起来是相当困难的，一般来说，我们认为表 2-15 所示情况下这种类型的值为空(null)。

表 2-15　空值情况

空值情况
没有设置为任何预定义的值
使用函数 unset()清除
明确地赋值为 null

例如，在以下代码中将空值赋给变量$a。

```
<?php
    $a = "我喜欢 PHP"; //为变量赋值非空值
     $a = null;//将变量值赋值为空,
     if(is_null($a)){
         echo "变量值是空值";

         }
?>
```

2.9　使用条件语句进行表单验证

条件语句用来判断某个表达式的结果是否满足规定的条件，然后执行相应语句。在流程控制中，条件语句的作用很强大。例如我们在检查用户名是否有效时，就可以通过条件语句进行控制。

 **视频教学：**光盘\videos\02\条件语句.avi　　　长度：7 分钟

2.9.1 基础知识——条件语句

条件语句是判断某个表达式的结果是否满足规定的条件，如果满足则执行该语句，如果不能满足就不执行该语句。在 PHP 中有很多用于判断的条件语句，如 if 语句、if...else 语句和 switch 语句等。

1. if…else 语句

在 PHP 中，有三种基本结构的 if…else 语句，这三种结构还可以进行嵌套使用。

1) 只有 if 语句

这种语句的语法如下：

```
if(表达式){
    语句体
}
```

当表达式为真时，执行语句体，否则跳过该语句体。如果语句体中只有一行，则可以省略大括号。例如：

```
if($a>$b){
    echo $a."大于".$b;
}
```

只有$a 的值大于$b 时，才能执行输出语句，否则将跳过该语句，继续执行下面的语句。

2) if…else 语句

这种语句的语法如下：

```
if(表达式){
  语句体 1
}else{
  语句体 2
}
```

当表达式为真时将执行语句体 1，否则将执行语句体 2。如果某个语句执行完毕后就会跳出整个结构。例如：

```
if($a<$b){
  echo $a."小于".$b;
}else{
  echo $a."是最大的";
}
```

2. switch 语句

要实现多分支选择结构可以有很多种方法，例如使用 if…else 语句，但是这种方法在只有一层的多分支结构中会显得十分冗长，此时可以使用 switch 语句使程序变得更加简单和明了。其语法如下：

```
switch(表达式){
   值 1:
```

```
    语句 1;break;
    值 2:
    语句 2;break;
    ...
    Default;
    语句 n;break;
}
```

当表达式的值为 1 时，执行语句 1 并且跳出该结构；如果值为 2 时，则执行语句 2 并且跳出；如果条件都不满足，则执行 default 中的语句。例如以下代码：

```
$b = 89;
switch(intval($b/10)){
    case 10:
    echo "成绩优秀 A+";break;
    case 9:
    echo "成绩 A";break;
    case 8:
    echo "成绩 A-";break;
    case 7:
    echo "成绩 B";break;
    case 6:
    echo "成绩 B-";break;
    case 5:
    echo "成绩不及格 C";break;
}
```



在 switch 语句中，对于初学者来说，最容易犯的错误就是漏掉每一个分支语句的 break 语句。如果没有 break 语句，switch 语句总是会检查每一个 case 分支，这样一来就降低了执行效率，还会出现不能预料的结果。

2.9.2　实例描述

进行表单处理时常常会用到流程控制语句，用得最多的就是使用条件语句进行对表单数据进行验证，下面通过一个实例来了解条件语句的使用。

2.9.3　实例应用

【例 2-7】 使用条件语句进行表单验证

(1)　我们使用第 2.7 节实例中所讲到的 index.html 页面，将表单中的 action="./register.php" 更改为新的表单处理文件 reg.php。

(2)　在 reg.php 文件中添加如下部分代码，并保存文件。

```
echo $username;
if($name=="wangyabin"){
    echo "该用户名已经被注册了<br>";
```

```
}else{
    echo "您注册的用户名有效<br>";
}
if($pass==$pass2){
    echo "你输入的密码一致";
}else{
    echo "您输入的两次密码不一致,请重新输入";
}
echo "<a href=Register1.html>返回</a><br>";
if($age<=18){
    echo "您今年".$age.",对不起,您还未成年,这种游戏不适合您! ";
}else{
   echo "恭喜您,游戏可以注册,但为了您的健康,请不要过度地玩游戏! ";
}
```

2.9.4 运行结果

将文件在浏览器中运行，可以看到注册页面如图 2-10 所示。



图 2-10 注册页面

在注册页面输入信息后单击【注册】按钮，就会跳转到 reg.php 页面进行验证，其结果如图 2-11 所示。

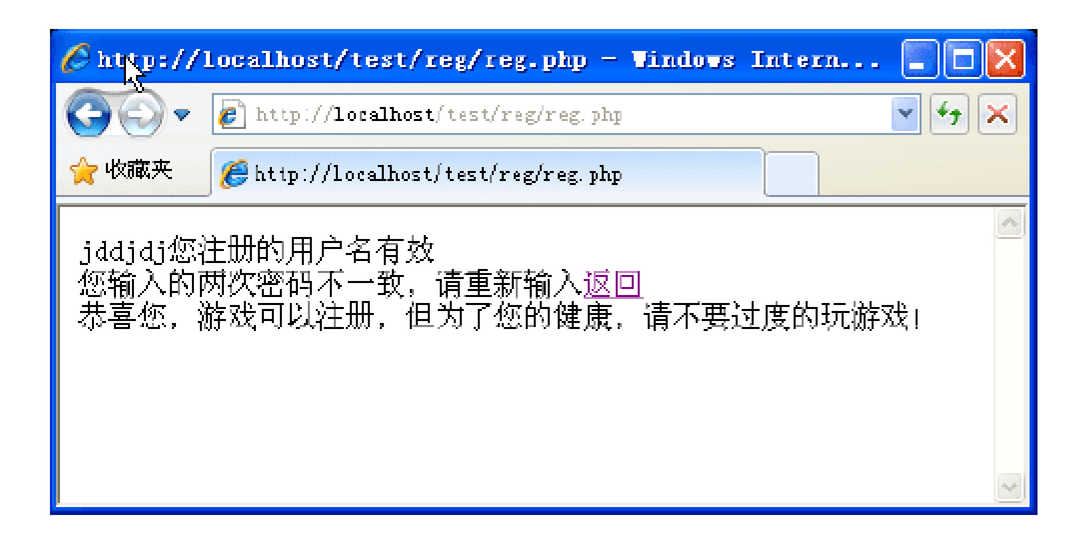


图 2-11 验证结果页面

2.9.5　实例分析



源码解析：

上述实例，在表单业务逻辑处理页面中我们使用了条件语句 if...else 对表单的输入数据进行了验证，并通过验证结果控制了程序的执行流程。

2.10　输出当前循环次数

循环语句是指让程序重复执行某个程序块，直到某个特定的表达式结果为假时，被执行的程序块才能结束，因此也被称为迭代语句。

视频教学： 光盘\videos\02\循环语句.avi　　长度：6 分钟
光盘\videos\02\While 循环语句.avi　　长度：5 分钟
光盘\videos\02\do while 语句.avi　　长度：5 分钟
光盘\videos\02\for each 循环语句.avi　　长度：6 分钟

2.10.1　基础知识——循环语句

在 PHP 中比较常见的循环语句有 for 循环、while 循环、do...while 循环和 foreach 循环。本小节将分别介绍这些循环语句。

1．for 循环语句

在 PHP 中，for 语句是功能最为强大的。任何的 while 和 do...while 循环都可用 for 循环来代替。在 for 循环中要先判断条件表达式是否为真，如果为真，则执行循环语句；如果为假，则不执行循环语句。其语法如下所示：

```
for(初始表达式 1,循环条件表达式 2,循环后的操作表达式 3){
    执行语句体;
}
```

其中表达式 1 在循环开始时被无条件地执行一次，然后开始执行表达式 2，如果为真，则执行表达式 3，然后再判断表达式 2 是否为真，为真则再次执行表达式 3，直到表达式 2 为假时，才会跳出循环。如以下例子：

```
for($s=1;$s<=5;$s++){
    echo "这是第".$s."个 PHPer<br>";
}
```

2．while 循环语句

在 while 循环语句中，表达式的值在每次的循环开始时就进行检查，如果为真则程序会向下执行循环，直到表达式的值为假时，跳出循环。只有当一开始表达式的值就为假时，循环语

句才一句都不会执行。其语法如下所示：

```
while(表达式){
    语句体
}
```

例如下面的代码循环输出从 3 到 9 的数字：

```
$in  = 3;
while($in<10){
    echo $in;
    echo "<br>";
    $in++;
}
```

3．do…while 循环语句

do…while 循环语句和 while 循环语句很相似，它是 while 循环语句的扩展，不同之处在于当循环中的表达式一开始就为假时，while 循环语句将不会执行，但 do…while 循环语句则至少执行一次。其语法如下所示：

```
do{
    语句体
}while(表达式)
```

例如下面的代码循环输从 0 到 10 的数字。

```
$itn = 0;
do{
    echo $itn;
    echo "<br>";
    $itn++;
}while($itn<=10)
```

4．foreach 循环语句

在 PHP 中，foreach 循环语句主要用于循环输出数组，这种循环语句可以迭代出集合中的每一项，但不能修改其中的任何一项。其语法如下所示。

```
foreach(集合 as 变量名){
    语句体
}
```

在 foreach 语句中，循环每执行一次，循环变量就可以从集合中读取一个元素。例如下面的代码：

```
$array = array("北京","上海","广州","深圳","香港");
 foreach($array as $ar){
    echo "我的城市是".$ar."<br>";
 }
```

在此例子中，我们可以看到，foreach 循环语句中的表达式是由两个参数组成的，用关键字

as 连接，关键字左边的是要访问的元素集合的名称，右边的是一个声明的临时变量。



任何一个循环语句都必须出现一次判断表达式的值为假的情况，否则循环因将形成死循环而无法结束。

2.10.2　实例描述

上次一个朋友问我，使用循环语句输出当前用户的 id 号，但是使用 for 循环的话，如果有 100 条信息，就会存在 100 次。那怎样只显示当前循环的次数和 id 呢。下面这个实例可以实现这个功能。

2.10.3　实例应用

【例 2-8】 输出当前循环次数

(1) 新建一个名为 cycle.php 的文件，在这里我们还没有连接数据库，所以以一个数组代替我们从数据库中查询的数据。其代码如下：

```
<?php
echo '正在处理的 ID:<span id=div_id></span>';//显示容器
$query=array(1,2,3,4,5,6,7,8,9,10);
for($i=0;$i<count($query);$i++){
$id=$query[$i];
echo "<script type=text/javascript>div_id.innerHTML='$id';</script>"; //用 JS
代码更新容器显示
flush();//刷新显示
}
?>
```

(2) 保存修改的文件，并放到服务器中。

2.10.4　运行结果

在服务器运行后，执行 cycle.php 文件，我们可以看到如下结果：

```
正在处理的 ID:10
```

2.10.5　实例分析



源码解析：

上述实例中，我们使用 for 循环输出从数据库中查询的数据，而且只输出一行数据。这样我们在做 Web 应用时，就显得安全性很高。

2.11 制作验证码随机数

在 PHP 中，函数分为系统内置函数和用户自定义函数，但不管是什么函数，所有的函数都具有同样的调用方法和取值。本节主要讲解函数的调用。

 视频教学：光盘/videos/02/函数及其调用.avi　　长度：5 分钟

2.11.1 基础知识——PHP 函数的调用

函数调用的语法形式为：

```
函数名(实际参数列表);
```

其中实际参数列表要与形式参数列表相对应(有默认参数时，实际参数还要与默认参数相对应)。如果实际参数比形式参数多，多余的参数会被自动舍弃；如果实际参数比形式参数少，实际参数会被填入形式参数中，不足的部分以空参数代替。下面我们来看一个实际调用函数的例子，代码如下：

```
echo rand(1,10);
```

在这段代码中，我们调用了 PHP 的内置函数 rand()来生成随机数。

2.11.2 实例描述

在进行 Web 开发时，我们经常会在注册或者登录的时候遇到要求输入验证码的情况，现代是网络时代，人们越来越注重网络的安全性。在网页中增加验证码的验证，无疑会给你的网页增加亮点。而验证码的随机数字我们有很多方法生成，下面通过一个实例来了解怎样使用 PHP 函数生成验证码的随机数。

2.11.3 实例应用

【例 2-9】 自作验证码随机数

(1) 新建一个名为 authimg.php 的文件，在文件中添加以下代码，这些代码用来生成验证码图片。

```
//生成验证码图片
Header("Content-type: image/PNG");
srand((double)microtime()*1000000);
$im = imagecreate(48,25);

//色彩设置
$r = Array(225,211,255,223);
```

```
$g = Array(225,236,237,215);
$b = Array(225,236,166,125);
$key = rand(0,3);
$bg = ImageColorAllocate($im, $r[$key],$g[$key],$b[$key]);
$stringColor = ImageColorAllocate($im, 255,51,153);
imagefill($im,68,30,$bg);
```

(2) 使用 PHP 的内置函数 rand()生成一个四位的随机数，并将四位整数绘入图片，代码如下：

```
//生成随机数
$ran=rand(1000,9999);
//将四位整数验证码绘入图片
imagestring($im, 5, 8, 6, $ran, $stringColor);
```

2.11.4　运行结果

在浏览器中运行文件后可以看到如图 2-12 所示的结果。

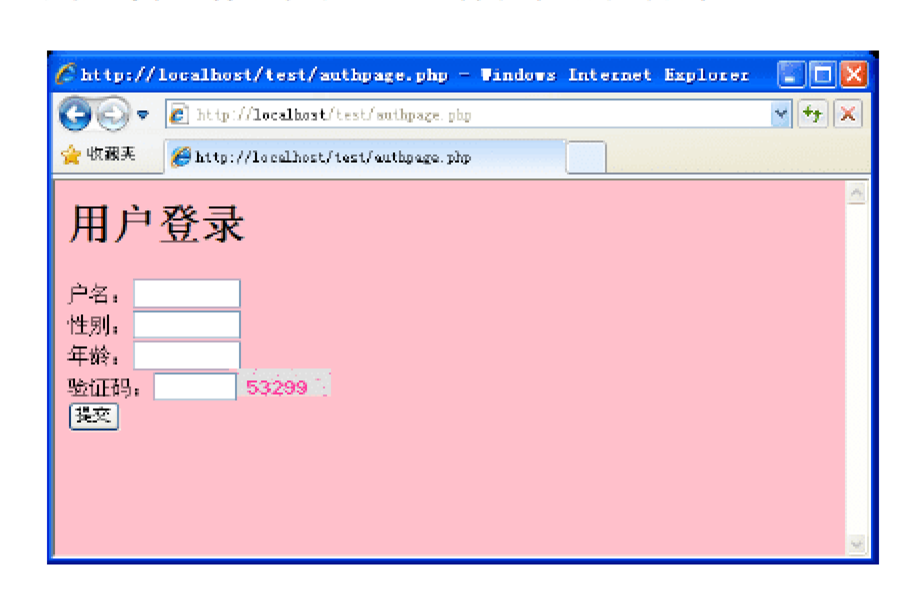


图 2-12　生成的随机数

2.11.5　实例分析



源码解析：

在上述实例中，我们使用 PHP 的内置函数 rand()来生成一个六位的随机数验证码，PHP 中还用许多功能强大的函数供我们使用。

2.12　自定义函数生成字符串验证码

PHP 本身就有大量的内置函数，这些函数能够使我们完成实际开发过程中的大部分工作。但是，还有一些比较个性化的要求，内置函数是不能完成的，这样就需要我们自己去定义函数以满足要求。



视频教学：光盘\videos\02\用户自定义函数.avi　　　　长度：14 分钟

2.12.1 基础知识——用户自定义函数

在 PHP 中，用户可以自己定义一个能够满足特定功能的函数，定义的语法格式如下：

```
function [&] function_name(参数){
     函数内的执行代码
}
```

函数中通常会返回一些值，在自定义函数中我们可以使用 return 关键字来返回相应的值，例如下面代码所示的例子：

```
function Area($width,$height,$type){
  //$type 的值为 1,为长方形
  if($type==1){
    return $height*$width;
  }
  //$type 的值为 2,为三角形
  if($type==2){
    return ($height*$width)/2;
  }
}
```

2.12.2 实例描述

从事 PHP 开发的都知道，虽说验证码可以增加网站的安全性，但安全性的高低则是取决于生成验证码随机数的难易，我们可以使用 PHP 的内置函数生成简单的随机数，也可以自己定义函数来生成更为复杂的验证码。

2.12.3 实例应用

【例 2-10】 制作验证码字符串

(1) 在 2.11 节的实例中，打开 authimg.php 文件，在文件中我们创建一个生成字符串的自定义函数，代码如下：

```
function randStr($len=6,$format='ALL') {
    switch($format) {
           case 'ALL':
          $chars='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
           break;
           case 'CHAR':
    $chars='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz';
           break;
    case 'NUMBER':
          $chars='0123456789';
    break;
         default :
```

```
            $chars='ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
            break;
     }
     $string="";
       //从字符源中随机取得字符
     while(strlen($string)<$len) {
          $string.=substr($chars,(mt_rand()%strlen($chars)),1);
    }
    return $string;
}
```

(2) 调用自定义函数，并生成验证码图片，代码如下：

```
//生成字符的个数
$randval = randStr(5,"");
//将验证码绘入图片
imagestring($im, 5, 8, 6, $randval, $stringColor);
```

(3) 我们使用 2.9 节的注册网页，在页面的表单中增加验证码图片代码，保存为 Registerl.php，如下所示：

```
<label for="yzm"> 输入验证码：</label>
<input class="input" type="text" id=authinput maxLength=15 style="width:
50px" name="authinput">
<img src=authimg.php?authnum=<? echo $authnum; ?>><br>
```

2.12.4　运行结果

将 Registerl.php 在服务器上运行后，可以看到如图 2-13 所示的结果。



图 2-13　生成字符串验证码

2.12.5 实例分析



源码解析：

上述实例中，我们自定义了 randStr($len=6,$format='ALL')函数，用来生成所需要的验证码，在自定义函数中，可以根据自己的需要来定义函数的功能。

2.13 实现验证码中的色彩设置

在制作验证码时，我们需要使用数组来控制色彩的设置，本节将介绍数组的相关知识。



视频教学：光盘\videos\02\创建数组.avi　　长度：7 分钟

2.13.1 基础知识——数组

数组就是将一些类型相同的数据聚集在一起，并按照特定的方式进行排列和引用。这种数据结构被称为数组(array)。在数组中每个索引对应一个值。索引可以是表示在数组中位置的数字，也可以是值与值之间的关联。具体的应用如以下代码所示：

```
<?php
   $name[0] = "王亚斌";
   $name[1] = "张俊驰";
   $name[2] = "王子轩";

   echo "我的名字叫：".$name[0]."<br>";
   echo "我的名字叫：".$name[1]."<br>";
   echo "我的名字叫：".$name[2];
?>
```

以上代码中，创建的数字是以数字进行的索引，按索引进行输出后其结果如下所示：

```
我的名字叫：王亚斌
我的名字叫：张骏驰
我的名字叫：王子轩
```

当然，也可以用值与值之间的关联进行索引，其代码如下：

```
$name[wangyabin] = "王亚斌";
$name[zhangjunchi] = "张俊驰";
$name[wangzixuan] = "王子轩";
```

按照上述的方法我们可以创建自己的数组，但在 PHP 中还可以使用 array()函数创建数组，其创建的示例代码如下：

```
<?php
    $name = array{"王亚斌","张俊驰","王子轩"};
    $name = array{
```

```
        'wangyabin'=>"王亚斌";
        'zhangjunchi'=>"张骏驰";
        'wangzixuan'=>"王子轩";
       }
?>
```



PHP 还支持多维数组。

2.13.2　实例描述

验证码中的色彩有背景色和字体颜色，这些颜色可以是固定的，也可以是随机的。我们可以先通过数组设置几种常用的基准色，然后再随机调用。

2.13.3　实例应用

【例 2-11】　实现验证码中的色彩设置

(1)　打开 2.11 节中的 authimg.php 文件的代码，根据颜色的搭配，我们可以在颜色基准色的数组中添加丰富的色彩，代码如下：

```
$r = Array(225,211,255,223,198);
$g = Array(225,236,237,215,139);
$b = Array(225,236,166,125,233);
```

(2)　修改背景颜色，代码如下：

```
$key = rand(0,4);
$bg = ImageColorAllocate($im, $r[$key],$g[$key],$b[$key]);
```

2.13.4　运行结果

运行页面后，可以看到如图 2-14 所示的效果。



图 2-14　生成彩色验证码图片背景

2.13.5 实例分析



源码解析：

上述实例，我们使用数组确定了 RGB 颜色设置时的数值，并在设置背景颜色时，随机调用，这样我们就可以通过数组设置不同的验证码背景色。

2.14 操 作 数 组

在 PHP 中，可以说数组的使用是非常频繁的，所以熟练地掌握对数组的操作，是 PHP 学习中一个非常重要的因素。下面我们来讲解怎样操作数组。

视频教学： 光盘\videos\02\使用 array().avi　　长度：5 分钟
光盘\videos\02\测试与输出数组.avi　　长度：5 分钟

2.14.1 基础知识——数组操作

在 PHP 中，操作数组主要是针对数组元素的增加、删除和定位数组元素，以及获取数组的大小。这些操作中我们需要用到相当多的数组函数，本节将详细地介绍。

1．数组的增加和删除

我们在进行 PHP 的开发时，对数组的增加和删除是经常使用的操作。一般我们通过 5 个函数来操作，如表 2-16 所示。

表 2-16　对数组增加和删除的函数

函数名称	作　用	语　法
array_push()	该函数将 variable 的值追加到 target_array 的末尾，成功时返回 true，否则返回 false。为此函数传递多个变量作为输入参数，并同时压入数组	int array_push(array target,miuxde variable)
array_pop()	该函数用于返回 target_array 的最后一个元素，并在结束后重置数组的指针	mixed array_pop(array targetrarray)
array_shift()	类似于 array_pop()，只是返回的是第一个数组的指针，结果所有相应的值会下移，并重置指针	mixed array_shift(array targetarray)
array_unshift()	类似于 array_push()，将元素增加在数组的头部，所有的元素会有新的位置，关联键不受影响	Int array_unshift(array target,miuxde variable)
array_pad()	该函数会修改 target 数组，将其大小增加到 length 指定的长度。通过由 pad_value 指定的值来实现。为正，填充到数组的右边；为负，填充到数组的左边	array_pad(array target,integer length,mixed pad_value)

2．定位数组元素

定位数组元素是指用户能通过 PHP 中的一些函数来对数组中的数据进行有效的筛选，下面我们来看看这些函数，如表 2-17 所示。

表 2-17　定位数组元素函数

函数名称	作　用	语　法
in_array()	该函数用于检测数组中是否存在某个值	Bool in_array(mixed needle,array haystack)
array_keys()	该函数用于返回数组中所有的键名，也就是返回一个由数组键组成的数组	Array array_keys(array target_array)
array_keys_exists()	该函数用于检查给定的键名或索引是否存在于数组中	Bool array_keys_exists(mixed key,array target_array)
array_values()	该函数用于返回数组中所有的值，并给其建立数字索引	array array_values(array target_array)
array_search()	该函数用于在数组中搜索给定的值，如果成功则返回相应的键名	Mixed array_search(mixed needle,array haystack)

3．确定数组大小及唯一性

在 PHP 中还有三个函数是用来确定数组中元素的总数和唯一值的个数的，如表 2-18 所示。

表 2-18　确定数组大小和唯一性函数

函数名称	作　用	语　法
count()	该函数用于返回数组中元素的总数	Int count(array target_array)
array_count_values()	该函数用于统计数组中所有的值出现的次数，返回一个包含关联键值对的数组	Array array_count_values(arrau arraytarget)
array_unique()	该函数会删除数组中所有重复的值，返回一个由唯一值组成的数组，该函数现将值作为字符串排序，然后对每一个值只保留第一个遇到的键名	Array array_unique(array targetarray)

2.14.2　实例描述

在 Web 开发中，我们经常会遇到一系列的报表，这些报表中含有大量的数据。在 PHP 中，从数据库中取出的这些数据都将被存放到数组中，因此，我们可以通过操作数组来操作这些数据，例如对这些数据进行排序等。

2.14.3　实例应用

【例 2-12】　数组排序

(1)　新建文件 2.14.php。在文件中，我们创建一个数组，其部分代码如下：

```
$people=array('name','sex','nation','birth');
```

```
foreach ($people as $mychrs)
echo $mychrs." ";
```

(2) 使用 sort()函数对数组进行升序排列，代码如下：

```
sort($people);
echo "<br />---排序后---<br />";
foreach ($people as $mychrs)
echo $mychrs." ";
```

2.14.4 运行结果

将文件 2.14.php 在浏览器中运行，我们可以看到排序前后的对比结果如下所示：

```
name sex nation birth
---排序后---
birth name nation sex
```

2.14.5 实例分析



源码解析：

上述实例中，我们使用 sort()函数对数组进行升序排序，并通过 foreach()函数对数组进行循环输出操作。

2.15 常见问题解答

2.15.1 用 for 循环显示最新数据问题



用 for 循环显示最新数据问题？

网络课堂：http://bbs.itzcn.com/thread-13629-1-1.html

想实现显示最新的三条数据，可编写的代码总是出错。我的代码如下：

```
$hd=$db->query("select id,title,hdtime,address,num_n,num_r,flag,jzbmtime,
bmnum,ifjh,picurl s from yzlove group club WHERE flag>0 order by id desc limit
0,3");
$rows = $db->fetch_array($hd);
$id = $rows[0];
```

上面的 SQL 语句有问题吗？还有要怎么使用 for 循环显示出结果？

【解决办法】：你的 SQL 语句没有问题。当执行到“$rows = $db->fetch_array($hd);”语句时$rows 是一个二维数组。此时可以像操作普通数组那样遍历它，即可得出结果。例如，下面的代码：

```
for($i=0;$i<count($rows);$i-++){
```

```
    echo $rows[$i]['title'];
}
```

2.15.2　PHP 变量声明的问题



PHP 变量声明的问题。

网络课堂：http://bbs.itzcn.com/thread-13630-1-1.html

请问在 PHP 中的变量和常量，他们有什么区别。主要是指他们原理上的区别，例如为什么会有变量，常量的作用又是什么？

【解决办法】：这个问题可谓是说来话长，还是给你总结一下吧。

(1) 首先要确定一点，PHP 是一门弱类型的语言。也就是说在使用变量时不用指定变量类型，在使用时也没有类型检查的一门编程语言。

(2) 基于弱类型的特点，所以在 PHP 中变量不用事先指定变量类型，而是在使用时根据程序由系统解释器来判断。

(3) 另外 PHP 的变量也有全局变量与局部变量之分。所谓全局变量指在程序运行期间都能使用的变量，而局部变量只在子函数或类中有效。

(4) 对于常量，它必须在使用前先定义，否则程序在执行时会出错。而且一旦定义了常量就不能被重新定义或者取消定义，其值也不能发生改变。但是变量的值可以随时发生改变，这也是常量与变量最根本的不同。

2.16　习　　题

一、填空题

(1) 获取数据类型使用函数__________；转换数据使用函数__________。

(2) 检查一个变量是否为空的函数是__________。

(3) PHP 变量名可以是__________开头。

二、选择题

(1) 使用 PHP 输出 "hello world"的是_________。

A. "Hello World";

B. echo"Hello World";

C. Document.Write("Hello World");

D. $a="Hello World";

(2) 使用"get"方法提交的表单中获取数据的是_________。

A. $_GET[];

B. Request.QueryString;

C. Request.Form;

D. $_POST[];

(3) 在 PHP 中创建函数的正确方法是________。

A. function myFunction()

B. create myFunction()

C. new_function myFunction()

D. new myFunction()

三、上机练习

上机练习： 自己动手制作登录页面，要求页面含有验证码。

练习要求：首先制作用于用户登录的页面，页面中不仅含有用户名、密码等基本信息，还要含有验证码，通过验证码进行验证后登录。执行效果如图 2-15 所示。

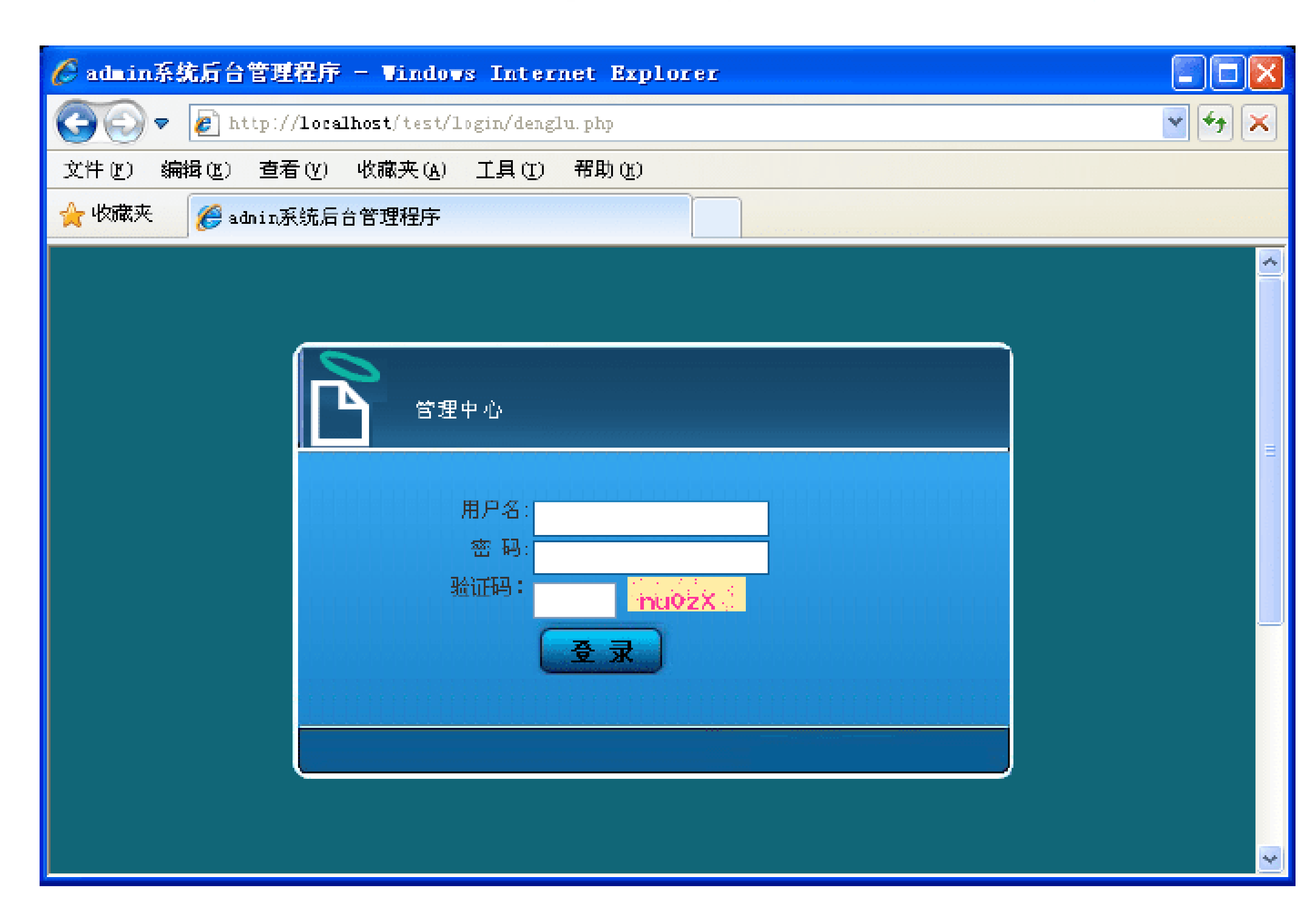


图 2-15　有验证码的登录页面



第 3 章　面向对象的 PHP

内容摘要：

PHP 4 已经提供了面向对象的功能，但它的面向对象功能太差了，PHP 5 完全重新开发了处理对象部分的内核，提供了更多功能的同时也提高了性能，例如面向对象的特性、类成员、构造函数、析构函数等，以及高级 OOP 的特性，如对象克隆、继承、抽象类、接口和反射等。本章将详细讲解 PHP 5 的一般面向对象特性和高级特性的应用。

学习目标：

- 熟悉 PHP 面向对象的特点。
- 灵活掌握类和对象。
- 熟练创建和使用字段、属性、常量和方法。
- 了解构造函数和析构函数。
- 了解新增 OOP 特性。
- 灵活运用 PHP 对象之间的克隆。
- 掌握类之间的继承。
- 掌握如何创建和使用接口。
- 掌握抽象类的创建和使用。
- 了解反射。

3.1 封装一个 Employee 类

面向对象是一种软件开发思想，它的诞生为开发策略范型带来了极大的改变，使编程的注意力从应用程序的逻辑回到其数据上来。利用它将数据和对数据的操作作为一个相互依赖不可分割的整体，即将对数据的操作作为一个对象进行处理，采用数据抽象和信息封装隐藏技术，这样就使应用程序更接近于我们周围的现实世界。以使软件开发简单化，符合人们的思维习惯，有助于控制软件的复杂性。

 **视频教学**：光盘\videos\03\OPP 特性.avi　　长度：5 分钟

3.1.1 基础知识——OOP 特性

面向对象编程语言更加接近物质世界。OOP 的内容主要涉及三大基础概念：封装、继承和多态。

1. 封装

封装就是隐藏类的数据成员，只向外提供一些公用的操作接口，用户只能通过这些接口来操作类的数据成员，而不能直接对这些数据成员进行赋值等操作。

2. 继承

继承是面向对象程序设计的主要特征之一，它可以让开发人员重用代码，可以节省程序设计的时间。继承就是在类之间建立一种相交关系，使得新定义的派生类的实例可以继承已有基类的特征和能力，而且可以加入新的特性或者是修改已有的特性，然后建立起新的层次。继承也可以是指一个对象从另一个对象中获得属性的过程，它支持按层次分类的概念。

继承与封装可以互相作用，提高软件模块的可复用性和可扩充性，还可以提高软件的开发效率，也能够利用前人或自己以前的开发成果。

3. 多态

多态是指同一操作作用于不同的对象，可以有不同的解释，产生不同的执行结果。其实，它的真正意义在于，在实际开发中只需要关注一个接口或基类的编程，而不必担心一个对象所属的具体类。

3.1.2 基础知识——类与对象

在现实生活中，每天我们都会接触到各种各样的实体，如人、动物、水果等，每个实体都由一组性质和行为来定义。例如，一只狗有毛皮颜色、腿的条数、品种等性质，并有会跑、会叫、吃食物等行为。实体的性质与行为的具体定义称为类(class)。

类实际上可以看作是一个实体模板，它定义了某个概念或真实事物的性质和行为。这个模板提供了一个基础，可以在此基础上创建实体的特定实例，这些特定的实例称为对象。对象是

类的概念实现，或者说是类实例化后的结果。如一只狗是动物类的一个实例化对象，狗具有动物的行为和特点。

在 OOP 中将每个实体的一组性质和行为分别称为字段和方法，它们定义了实体的形态。在 PHP 中，一般类的创建语法如下所示：

```
class classname
{
    //声明字段
    //声明方法
}
```

在上述代码中，classname 是指类名，一个类由成员方法和成员字段组成。

3.1.3　实例描述

曾经使用 PHP 开发过一个人事管理网站，当时就是通过面向对象思想封装了一个 Employee 类，当实例化一个类对象时，就在程序中形象地刻画出了企业的每一个员工，这样就可以在程序中对每个员工进行管理了。本节实例就具体讲解如何封装一个 Employee 类。

3.1.4　实例应用

【例 3-1】 封装一个 Employee 类

新建一个 employee.php 文件，根据 OOP 的特性，封装一个 Employee 类，代码如下所示：

```
<?php
    class Employee
    {
        public $name;
        public $sex;
        public $age;
        public $email;
        function NewEmployee($p_name,$p_sex,$p_age,$p_email)
        {
            $this->name = $p_name;
            $this->sex = $p_sex;
            $this->age = $p_age;
            $this->email = $p_email;
            printf("员工姓名：%1\$s<br/>性别：%2\$s<br/>年龄：%3\$s<br/>电子邮
件：%4\$s<br/>", $this->name,$this->sex,$this->age,$this->email);
            if($this->sex=="女")
            {
                echo "欢迎".$this->name."，加入本社团！";
            }
            else
            {
                echo "对不起，本社团仅限女性加入！" ;
```

```
            }
        }
    }
    $employee = new Employee();
    $employee->NewEmployee("胡青青","女",25,"huqingqing@sina.cn");
?>
```

3.1.5 运行结果

打开 IE 浏览器在地址栏中输入“http://localhost/PHP_3/employee.php”，执行效果如图 3-1 所示。

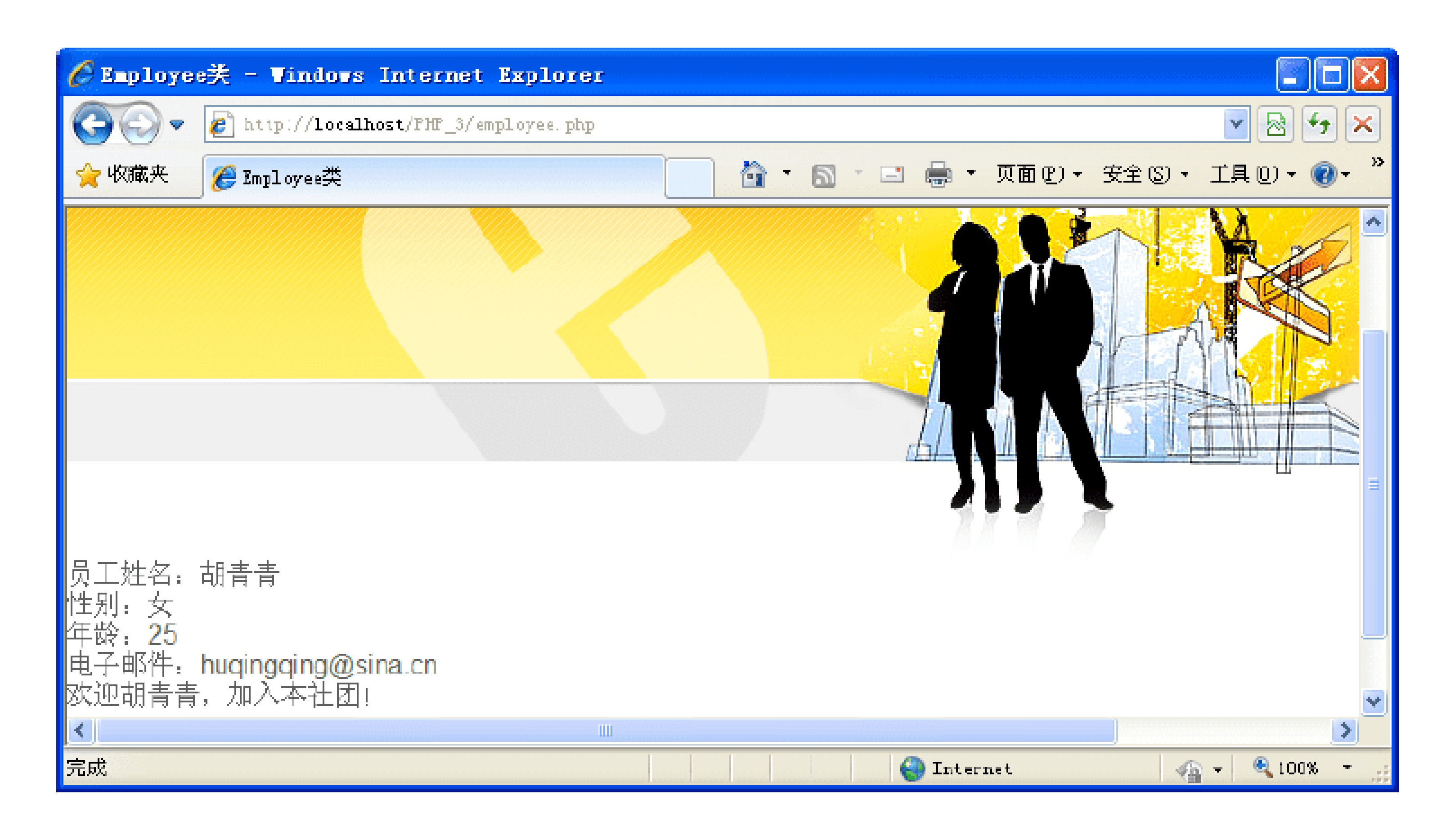


图 3-1 Employee 类

3.1.6 实例分析


源码解析：

定义一个 Employee 类，并为该类声明了 4 个字段，分别是 name、sex、age 和 email。该类还包括一个申请加入社团的函数 NewEmployee()。在 PHP 中，创建一个实体对象，和在其他一些面向对象语言一样，需要使用 new 关键字来创建。创建对象完成之后可以通过对象变量来调用类中的函数和字段。

3.2 创建用户类

类用于表示要在应用程序中处理的实际事物。例如，假设要开发一个人事管理系统，可能就要包括一些用来表示员工、任务、部门等的类。每个实体都包含一组性质和行为，在 OOP 中分别称为字段和方法，它们定义了实体的形态。


视频教学：光盘\videos\03\字段.avi　　长度：11 分钟

光盘\videos\03\属性.avi　　长度：9 分钟
光盘\videos\03\自定义属性.avi　　长度：5 分钟
光盘\videos\03\常量.avi　　长度：8 分钟
光盘\videos\03\方法.avi　　长度：15 分钟
光盘\videos\03\抽象方法.avi　　长度：7 分钟

3.2.1　基础知识——类的组成

类主要由字段、属性、常量与方法组成。本节主要讲解类的组成元素字段、属性、常量与方法的声明和使用。

1. 字段

字段用于描述类的各方面性质。从表面上看它与一般的 PHP 变量非常相似，但还有一些细微的差别。本节将详细介绍如何声明字段、使用字段及类中字段的作用范围。

1)　声明字段

类中定义的字段，实际上指定的就是对象中所具有的属性，它表示的是实体的一种性质与状态。字段的声明与变量声明的规则非常相似，不过还有一点区别就是创建的位置不同。PHP 是一种弱类型语言，字段甚至不需要声明，就可以直接创建并赋值，但是一般很少这样使用。通常采用在类的开始处声明字段，此时也可以为字段赋初值。代码如下所示：

```
class Customer
{
    public $c_name = "张小华";        //声明并赋值
    private $c_price;
}
```

上述代码中，定义了两个字段$c_name 和$c_price，public 和 private 表示这个字段的作用域范围，并为变量$c_name 赋了初始值，$c_price 没有赋初值。字段被指定了作用域范围就必须在指定范围内使用。

2)　使用字段

字段的使用与变量不同，不能直接使用，需要通过“->”操作符引用，并且只能使用对象调用，所以需要指定相应的对象，才能使用字段，使用字段的语法格式如下所示：

```
Object->field
```

在类的内部使用类字段，也需要使用“->”操作符，但此时不使用相应的类名，而是使用$this 关键字，表示引用当前类中的字段。这一点与 Java 语言中使用 this 关键字引用成员变量相同。下面通过定义一个 Employee 类来演示如何在类中与类外引用字段的示例，代码如下所示：

```
<?php

    class Employee
    {
```

```
        public $name;
        public $salary;
        public $age;
        function working(){
            printf("%1\$s 的薪水是：%2\$s",$this->name,$this->salary);
        }
    }
    $emp = new Employee();
    $emp->name = "杨华强";
    $emp->salary = 5000;
    $emp->age = 26;
    $emp->working();
?>
```

上述代码中，定义了一个 Employee 类，在该类中声明了 3 个字段，分别是员工姓名$name、薪水$salary 和员工年龄$age，还有一个 working()函数。类定义完成之后，通过 new 关键字实例化了一个对象$emp，利用该对象通过操作符“->”为对象中的字段赋值，并调用函数 working()，在页面中输出显示该员工的薪水情况，执行效果如下所示：

```
杨华强的薪水是：5000
```

3) 字段的作用域

PHP 为字段提供了 5 个作用域范围，分别为 public、private、protected、final 和 static(static 作用域将在后面章节中讲解)，使用这些作用域可以影响类字段的可用空间。5 个作用域的有关介绍如表 3-1 所示。

表 3-1 访问控制修饰符

作 用 域	说明描述
public	表示修饰的字段为公共的，在任何地方都可以访问到，但在 OOP 中。为避免数据的安全性，并不鼓励使用公共字段
private	表示修饰的字段为私有的，不能由实例化的对象直接访问，也不能由其子类使用，但可以在类中访问，类以外的其他位置也无法访问
protected	表示修饰的字段为受保护的。受保护的字段与私有字段有些相似，都只能在类内部使用，此外还可以在继承的子类中访问和操作受保护的字段，这一点是私有字段没有的特性
final	表示修饰的字段为不可在子类中覆盖的
static	静态作用域，在以后章节详细讲解

2. 属性

在 PHP 5 中没有提供像其他 OOP 语言一样处理并使用属性的功能，因此需要我们自己使用公共方法来模拟实现这一功能。例如：声明两个公共方法 getName()和 setName()，为属性 name 分别创建获取方法和设置方法，并在方法中添加一些适当的语句实现功能。

另外还有一种方法，就是可以重载__set()方法和__get()方法来实现对属性的处理。当在 PHP

类代码中应用一个类没有定义的成员变量时，将会自动触发执行__set()或__get()方法。属性的用途有很多，如：产生错误消息，通过动态创建新的变量来扩展类等。本节将详细介绍__set()方法和__get()方法，以及创建自定义获取方法和设置方法。

1)　__set()

__set()方法可以用于为隐藏的字段赋值，并且还可以在为类字段赋值之前添加一些验证类数据的代码。可以为该方法传递两个参数，分别是属性名和相应的值。如果方法执行成功将返回 true，否则将返回 false。该方法的语法格式如下所示：

```
boolean __set([String property_name,][mixed value_to_assign])
```



__set()方法前面是两个下划线，而不是一个，如果格式不正确，该方法将不能执行。

2)　__get()

_get()方法可以用于获取类变量的值，只需要向该方法传递要获取的类变量名称即可。如果方法执行成功将返回 true，否则将返回 false。该方法也可称为访问方法(accessor)或获取方法(getter)。它的语法格式如下所示：

```
boolean __get([string property_name])
```

3)　创建自定义获取方法和设置方法

上面讲解__set()方法和__get()方法，结合使用这两个方法可以实现对属性的处理。但是这种方式无法处理大型且比较复杂的面向对象应用程序中的属性。因为 PHP 不支持像 Java 或 C#的方式来创建属性，所以需要在为类创建一个变量时，再为该变量创建设置变量值和获取变量值的方法。下面通过一个示例来理解如何创建自定义获取和设置变量值的方法，代码如下所示：

```
<?php
    class User
    {
        private $username;
        public function getUsername()
        {
            return $this->username;
        }
        public function setUsername()
        {
            $this->username = $username;
        }
    }
    $user = new User();
    $user->setUsername("小鱼儿");
    echo ($user->getUsername());
?>
```

上述代码创建了一个 User 类，声明了一个$username 变量，并为该变量创建了自定义的获取和设置变量值的方法，分别是 getUsername()和 setUsername()。最后实例化了一个 User 类对象$user，通过对象调用 getUsername()和 setUsername()函数来获取和设置变量值。采用这种方

式没有使用属性那么灵活方便，但确实通过使用标准的命名约定封装了管理和存取任务。当然，还可以向设置方法中添加另外的验证功能。

3. 常量

在 PHP 5 中可以使用 const 修饰常量。常量在定义好之后，其值不能在类中发生改变。

另外使用 const 关键字定义的常量与定义变量的方法不同，常量不需要加$修饰符，而且常量名称一般都大写，这是一个约定，在任何语言中都是这样。在类中定义常量语法格式如下所示：

```
const NAME = 'VALUE';
```

类常量的访问方式与其他普通变量不同，常量值不能使用对象的实例来访问，而是使用$object::constant 表达式来访问。



常量值不能是一个变量、类的成员、数学表达式或函数调用的结果，它必须是一个常量表达式。

4. 方法

方法与函数非常相似，有一点不同就是方法是用来定义类的行为，而函数定义比较随便，方法与函数最突出的共同点就是都可以接受参数，并向调用者返回一个值。

在 PHP 脚本中，需要调用一个函数时，直接通过函数名即可实现，但调用类的方法则需要通过对象名称来调用。调用方法的语法格式如下所示：

```
Object->method_name();
```

1) 声明方法

定义一个类方法的语法格式和函数差不多一样，唯一的区别就是定义类中的方法需要使用 public、protected 或 private 等作用域。如果没有设置作用域，默认作用域为 public。声明方法的语法格式如下所示：

```
scope function functionName()
{
    //方法体
}
```

上述语法格式中，scope 表示方法的作用域范围，functionName 表示方法的名称，大括号中的内容表示方法体，即当调用该方法时将要执行的代码。下面创建一个空调类 Conditioner，并定义一个制冷功能的方法 cold()，代码如下所示：

```
<?php
    class Conditioner{
        private $style = '立式空调';
        public function cold(){
            echo "空调样式为：".$this->style.",它可以制冷！";
        }
    }
?>
```

在上述代码的 Conditioner 类中定义了一个$style 字段和一个 cold()方法，在该方法中输出空调的样式和具有制冷功能。由此可以得出，类中的方法可以操作类本身的字段。实际上，方法也可以像函数一样传入一些参数，或向调用者返回一个值等。



声明公共方法，可以不显式地声明作用域，而是像声明函数那样(不添加任何作用域)声明方法，因为方法的默认作用域就是公共的。

2) 调用方法

调用方法与调用函数几乎相同，下面通过一个示例来看看如何调用方法，代码如下所示：

```
<?php
    class Count
    {
        private $num1=40;
        private $num2=50;
        public function add(){
            return $this->num1+$this->num2;
        }
    }
    $c = new Count();
    echo "计算两个变量的和为：".$c->add();
?>
```

上述代码创建了一个 Count 类，在该类中声明了两个变量$num1 和$num2，还有一个 add()方法，用来计算$num1 和$num2 的和，并将计算结果返回。最后实例化 Count 类得到一个对象$c，使用类对象$c 调用 add()方法，并输出两个变量的和。执行上述代码后，结果如下所示：

```
计算两个变量的和为：90
```

3)　方法作用域

PHP 提供了 6 种方法作用域，它们分别是 public、private、protected、abstract、final 和 static。本节主要讲解前 5 种作用域，static 作用域将在后面章节进行讲解。表 3-2 简要介绍方法作用域。

表 3-2　方法作用域

关 键 字	说　明
public	表示方法是一个公共方法，公共方法与公共变量一样，可以在 PHP 脚本中的任何地方通过对象名来访问
private	表示方法是私有的，声明为私有的方法只能在类的内部使用，不能由类的实例化对象调用，也不能由类的子类调用。如果某些方法只用作类中其他方法的“助手”(即辅助方法)，就应当声明为私有的
protected	表示方法是一个受保护的方法，声明为受保护的方法，只能在类的内部和类的子类中使用
abstract	表示方法是一个抽象方法。抽象方法很特殊，只具有方法名，而没有方法体的方法，它的具体实现需要在子类中完成
final	表示最终的方法，声明为 final 的方法不可以被子类覆盖

3.2.2 实例描述

用户注册一个网站账号，当用户输入信息之后，将信息数据提交给后台，后台在获取用户信息之前，还需对用户的信息进行校验，可以将用户信息作为参数传递到类的实例化方法(即构造函数中)进行验证。要实现合理的验证，就要封装一个完美的用户类，本节实例就封装了一个相当不错的用户类，来看看吧。

3.2.3 实例应用

【例 3-2】 创建用户类

新建一个 user.php 文件，在该文件中封装定义一个用户类，包含用户的基本信息字段，如用户名、密码等。代码如下所示：

```
<?php
    class User{
        private $username;
        private $password;
        function __construct($username,$password)
        {
            if($this->test($username,$password))
            {
                echo "用户".$username."通过验证,注册成功!! <br/>";
            }
            else{
                echo "对不起，您输入了不合法数据,请核实重新输入!! <br/>";
            }
        }

        protected function test($name,$pwd)
        {
            if($name==""||$pwd=="")
            {
                    return false;
            }
            else
            {
                return true;
            }
        }
    }
    $user = new User("xiaoyuer","123456");
    $u = new User();
?>
```

3.2.4　运行结果

打开 IE 浏览器，在地址栏中输入“http://localhost/PHP_3/user/user.php”，执行效果如图 3-2 所示。



图 3-2　用户注册验证

3.2.5　实例分析



源码解析：

本实例定义了一个 User 类，为该类声明两个字段，分别是 username 和 password。此外，定义了一个构造函数，在构造函数中对初始化对象传入的参数值进行校验之后，再为字段赋值。这样可以避免将用户输入的不合法数据引入程序中。

3.3　构造一个演员

现实世界中，任何一个物体都有一个由新生到消亡的过程。在应用程序中，实例化的类对象也有这个过程，可以通过构造函数创建出一个类对象，也可以通过析构函数销毁一个类对象。本节主要介绍构造函数和析构函数。

视频教学： 光盘\videos\03\构造函数.avi　　长度：14 分钟

光盘\videos\03\析构函数.avi　　长度：5 分钟

3.3.1 基础知识——构造函数

大多数类中都有一种称为构造函数的特殊方法。当创建一个对象时，它将自动调用构造函数，也就是使用 new 这个关键字来实例化对象的时候自动调用构造方法。

构造函数的声明与其他函数的声明一样，只是其名称必须是__construct()。这是 PHP 5 中的变化，以前的版本中，构造函数的名称必须与类名相同，这种方式在 PHP 5 中仍然可以使用，但现在已经很少有人用了，这样做的好处是可以使构造函数独立于类名，当类名发生改变时不需要改相应的构造函数名称了。为了向下兼容，如果一个类中没有名为__construct()的方法，PHP 将搜索一个 PHP 4 中的写法，即与类名称相同名称的构造函数。定义一个构造函数的语法格式如下所示：

```
function __construct([arg1,arg2,...,argn])
{
    //方法体
}
```

另外，还可以在构造函数中调用类本身的方法和其他类的构造函数，调用类本身的方法通过$this->方法名即可调用，调用其他类构造函数的语法格式如下所示：

```
classname::__construct();
```

3.3.2 基础知识——析构函数

析构函数可以用来关闭数据库连接、记录调试信息等一些清除收尾的工作。当指向这个对象的最后一个引用被销毁时，析构方法被调用，调用完成后释放内存，并且析构函数不接受任何参数。在 PHP 4 以前的版本中没有析构函数，尽管脚本执行结束后也会自动销毁，但是不太可能自定义这个清除过程，PHP 5 出现后引入了析构函数，就可以自定义了。

创建析构函数与创建其他方法一样，但是析构函数必须命名为__destruct()。下面定义一个简单的析构函数，代码如下所示：

```
function __destuct(){
    print("释放内存中的所有对象");
}
```

3.3.3 实例描述

现在随着网络的发展，各种信息的传播速度更快捷了。一些演员为了提高自己的人气，给自己吸引来更多的粉丝，纷纷给出了自己的博客或个人空间等。本节实例通过定义一个 Role 类，封装演员的基本信息，最后构造出一个演员实体对象，制作演员的特色空间。

3.3.4 实例应用

【例 3-3】 构造一个演员

创建一个带有构造函数的 Role 类，并为该类声明了演员姓名 name、性别 sex、年龄 age 和人气指数 fame，代码如下所示：

```
<?php
    class Role
    {
        private $name;      //演员姓名
        private $sex;       //性别
        private $age;       //年龄
        private $fame;      //人气指数
        function __construct($name,$sex,$age,$fame)    //构造函数
        {
            $this->setName($name);
            $this->setSex($sex);
            $this->setAge($age);
            $this->setFame($fame);
        }
        function setName($name)
        {
            $this->name = $name;
            echo "演员姓名: ".$this->name."<br/>";
        }
        function setSex($sex)
        {
            $this->sex = $sex;
            echo "性别: ".$this->sex."<br/>";
        }
        function setAge($age)
        {
            $this->age = $age;
            echo "年龄: ".$this->age."<br/>";
        }

        function setFame($fame){
            $this->fame=$fame;
            echo "人气指数: ".$this->fame."<br/>";
        }
    }
    $role = new Role("加菲猫","男",4,"******");
?>
```

3.3.5　运行结果

打开 IE 浏览器，在地址栏中输入“http://localhost/PHP_3/role.php”，执行效果如图 3-3

所示。



图 3-3　演员加菲猫

3.3.6　实例分析



源码解析：

Role 类声明了 4 个字段，分别是$name、$sex、$age 和$fame。使用构造函数统一标识__construct 为类 Role 定义了一个包含 4 个参数的构造函数，还定义了设置各个字段值的函数，如 setName()、setSex()、setAge()等。实例化类对象可以直接传给构造函数参数，来初始化字段值。

3.4　新增 OOP 特性

在 PHP 5 中添加了许多 OOP 特性，本节主要讲解这些新增的 OOP 特性，例如类型提示、instanceof 关键字和自动加载对象等。

视频教学： 光盘\videos\03\类型提示.avi　　长度：6 分钟
光盘\videos\03\静态成员.avi　　长度：9 分钟
光盘\videos\03\instanceof.avi　　长度：6 分钟
光盘\videos\03\自动加载对象.avi　　长度：10 分钟

3.4.1　基础知识——类型提示

类型提示可以确保传递给方法的对象确实是所期望的类实例。例如，将类 Employee 的对象传递给 work(Employee $employee)方法才有意义。因此，可以在方法定义的唯一输入参数$employee 前面加上 Employee，强制实施此规则，代码如下所示：

```
public function take_lunchbreak(Employee $employee)
{
    ...
}
```



类型提示只用于对象，而无法为整数、浮点数或字符串等类型提供提示。

3.4.2　基础知识——静态类成员

静态类成员与一般的类成员有一些区别，静态成员只与类本身有关，与实例对象没有关系。静态类成员指的是静态属性和静态方法，通过在类中声明静态属性和静态方法可以封装定义类的数据和功能。一般类成员可以用来定义类特定对象的功能和数据。

当需要为类声明静态成员时，在方法或字段前面添加 static 关键字即可。类的静态成员对于类的所有实例化对象都是可见的，任何一个实例化对象修改了静态类成员都会反映到所有实例化对象中。

下面来看一个包含静态类成员的类 Counter，统计创建的类对象个数，代码如下所示：

```
<?php
    class Counter
    {
        private static $count = 0;
        function __construct()
        {
            self::$count++;
        }
        static function getCount()
        {
            return self:$count;
        }
    }
    $c1 = new Counter();
    echo "创建了第".Counter::getCount()."个对象<br/>";
    $c2 = new Counter();
    echo "创建了第".Counter::getCount()."个对象<br/>";
?>
```

3.4.3　基础知识——instanceof 关键字

PHP 5 另一个新成员是 instanceof 关键字。使用这个关键字可以用来确定一个对象是一个类的实例、一个类的子类，还是实现了某个特定接口。下面通过一个小实例来演示该关键字的使用，代码如下所示：

```
<?php
    class Test{}
    $t = new Test();
```

```
    if($t instanceof Test)      //判断$t 是否是类 Test 的实例
    {
        echo "Yes";
    }else{
        echo "No";
    }
?>
```

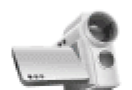
3.4.4 基础知识——自动加载对象

在实际项目开发中，有时可能会需要引入一些第三方类库，PHP 5 中也提供了类似的功能，引入了自动加载对象的概念，可以在项目中引入一些现成的类库，已达到代码重复利用，从而提高开发效率，节约开发成本。需要在脚本中使用一些脚本未定义的类库或函数，可以使用 require_once()语句实现。例如，在页面中引入 teacher 类，需要插入下面的语句：

```
require_once("myweb/teacher.class.php");
```

3.5 图 书 管 理

继承是 OOP 一个非常重要的特性，它使类和类之间有了相应的层次结构，使类的管理更加清晰，并且提高了代码的可重用性。PHP 中的继承类称为派生类或子类，被继承类称为基类或父类。本节将详细介绍继承的应用。

 **视频教学**：光盘\videos\03\继承.avi　　　　长度：11 分钟

3.5.1 基础知识——继承

类继承可以通过 extends 关键字实现，并且 PHP 中的类继承和其他面向对象语言一样也属于单继承，即一个类只能有一个基类，但是一个类可以被多个子类继承，继承语法格式如下：

```
class Son extends Father
{
    //类成员
}
```

子类可以从基类中继承所有公共成员，当然包括构造函数，在子类中运用基类中的构造函数与运用其他成员的不同之处在于子类可以显式调用基类中的构造函数，也可以隐式调用基类中的构造函数。如果父类中有构造函数，并且子类中没有构造函数，那么子类在实例化时执行父类的构造函数。代码如下所示：

```
<?php
    class Father
    {
        function __construct()
```

```
        {
            echo "父类产生了.<br/>";
        }
    }
    class Son extends Father
    {
        function study()
        {
            echo "子类方法.";
        }
    }
    $son = new Son();
    $son->study();
?>
```

上述代码执行效果如下所示：

```
父类产生了.
子类方法.
```

一个类可以调用另外一个类的构造函数，即使这两个类不是基类与子类关系也可以。非继承关系调用构造函数称为显式调用。如果一个子类继承了一个父类，子类只需要使用关键字 parent 就可以直接调用父类的构造函数，称为隐式调用。

下面分别演示如何显式与隐式调用父类的构造函数，代码如下所示：

```
<?php
    class Base
    {
        function __construct(){
            print "父类由此产生<br/>";
        }
    }
    class Sub extends Base{
        function __construct()
        {
            //显示调用父类构造函数
            Base::__construct();
            //隐式调用父类的构造函数
            parent::__construct();
            echo "执行子类本身的构造函数<br/>";
        }
    }
?>
```

3.5.2　实例描述

刚做了一个图书管理系统，在该系统中需要对图书进行分类管理，还需要将热销图书摆放到系统首页货架上，因此抽象出一个图书模板类，包含图书的一些基本信息，然后再定义一个

子类继承图书模板类，并为子类添加一个特有属性——图书类别属性，这样就可以对图书进行分类管理了，还可以展示新书推荐，最后还定义了一个热销图书类，该类也继承图书模板类，并重写父类的一个显示图书的方法，将热销图书放到首页货架上，这样一切操作更清晰明了，这就是灵活使用继承的好处。下面来看看实例的具体实现吧。

3.5.3 实例应用

【例 3-4】 图书管理

(1) 新建一个 book.php 文件，在文件中首先定义一个图书模板类，包含了图书名称、价格、数量和折扣等基本信息，代码如下所示：

```
/*  父类  */
class BookObject{
    public $object_name;                                  //图书名称
    public $object_price;                                 //图书价格
    public $object_num;                                   //图书数量
    public $object_agio;                                  //图书折扣
    function __construct($name,$price,$num,$agio){        //构造函数
        $this -> object_name = $name;
        $this -> object_price = $price;
        $this -> object_num = $num;
        $this -> object_agio = $agio;
    }
    function showMe(){                                    //输出函数
        echo '显示图书信息';
    }
}
```

(2) 定义一个 Book 类，继承 BookObject 类，为该类添加一个新属性，并重写 showMe()方法，代码如下所示：

```
/*  子类 Book  */
class Book extends BookObject{                        //BookObject 的子类
    public $book_type;                                //类别
    function __construct($type,$num){                 //声明构造方法
        $this -> book_type = $type;
        $this -> object_num = $num;
    }
    function showMe(){                                //重写父类中的 showMe 方法
        return '本次新进'.$this -> book_type.'图书'.$this->object_num.'本<br>';
    }
}
```

(3) 定义一个热卖图书类 Elec，继承 BookObject 类，该类重写父类的 showMe()方法，代码如下所示：

```
/*  子类 Elec  */
class Elec extends BookObject{                        //BookObject 的另一个子类
```

```
    function showMe(){                          //重写父类中的 showMe 方法
        return '热卖图书：'.$this -> object_name.'<br>原价：'
        .$this -> object_price.'<br>特价：'.$this -> object_price * $this -> 
object_agio;
    }
}
```

(4) 创建 Book 对象和 Elec 对象，并分别调用 showMe()方法，显示新进图书与热销图书。代码如下所示：

```
$c_book = new Book('计算机类',1000);            //声明一个 Book 子类对象
$h_elec = new Elec('PHP 函数参考大全',98,3,0.8); //声明一个 Elec 子类对象
echo $c_book->showMe()."<br>";                  //输出 Book 子类的 showMe()方法
echo $h_elec->showMe();                         //输出 Elec 子类的是 showMe()方法
```

3.5.4　运行结果

打开 IE 浏览器在地址栏中输入“http://localhost/PHP_3/book.php”，执行效果如图 3-4 所示。



图 3-4　图书管理

3.5.5　实例分析


源码解析：

本实例首先定义了一个图书模板类 BookObject，在该类中定义了一个图书的基本信息，然后定义了一个图书类 Book，为该类添加了一个图书类型属性 book_type，并重写了 showMe()方法，显示新书信息。最后定义了一个 Elec 类，在该类中重写了 showMe()方法，显示热销图书的信息。

3.6 输出日志信息

实际上接口与类非常相似，只不过类中的成员定义了详细的实现方法，而接口只包含成员的声明，没有构造函数、字段等，并且也不能有实例。本节将详细介绍接口的创建与实现。

 **视频教学：**光盘\videos\03\接口.avi　　长度：10 分钟

3.6.1 基础知识——接口

使用接口可以定义某种服务的一般规范，声明了所需的函数和常量，但不定义出实现过程，仅是一个函数和常量的声明。之所以不给出实现的细节，是因为不同的实体可能需要用不同的方式来实现公共的方法定义。关键是要建立必须实现的一组一般原则，只有满足了这些原则才能说实现了这个接口。

在接口中所有的方法必须声明为 public，这是接口的特性。创建一个接口需要使用关键字 interface，创建接口的语法格式如下所示：

```
interface InterfaceName
{
    const name;
    function methodName();
}
```

定义了一个接口，但没有相关的类去实现，该接口是毫无意义的。一个类继承实现接口需要使用 implements 关键字，该关键字表示实现一个接口。在接口中所有的方法必须在实现类的内部实现，疏忽这些将导致一个致命的错误。

类也可以实现多个接口，通过逗号隔开。接口和接口之间也可以进行继承，就像类与类之间的继承一样，也是通过 extends 关键字实现。下面就来看一个完整的接口应用例子，代码如下所示：

```
<?php
    interface Live
    {
        public function eat();
    }
    interface Animal extends Live       //继承 Live 接口
    {
        public function sleeping();
        public function running();
    }
    class Dog implements Animal         //实现 Animal 接口
    {
        public function eat(){
            echo "可以吃东西了！<br/>";
```

```
        }
        public function sleeping()
        {
            echo "也可以睡觉了！<br/>";
        }
        public function running(){
            echo "还可以跑,非一般的感觉呀！<br/>";
        }
    }
    $dog = new Dog();
    $dog->eat();
    $dog->sleeping();
    $dog->running();
?>
```

3.6.2 实例描述

前段时间和以前的老同学一起做了个统计用户操作信息的系统，该系统的核心就是定义了一个日志接口，让需要追踪记录的操作实体来实现日志接口。下面通过本实例演示接口的具体应用。

3.6.3 实例应用

【例 3-5】 输出日志信息

(1) 新建一个 interface.php 文件，在该文件中定义了一个日志接口，声明了一个输出日志信息的方法 logString()，代码如下所示：

```
<?php
interface Loggable
{
    function logString();
}
?>
```

(2) 新建一个 Person 类，该类实现了 Loggable 接口，给出了 logString()方法的具体实现代码，将在日志中输出该用户的基本信息，代码如下所示。

```
<?php
class Person implements Loggable
{
    private $name;
    private $address;
    private $idNumber;
    private $age;
    function __construct($name,$address,$idNumber,$age)
    {
        $this->name=$name;
```

```
        $this->address=$address;
        $this->idNumber=$idNumber;
        $this->age = $age;
    }
    function logString()
    {
        printf("用户名称: %1\$s  编号: %2\$s  年龄: %3\$s
            家庭住址: %4\$s  <br/>",$this->name,$this
            ->idNumber,$this->age,$this->address);
    }
}
?>
```

(3) 新建了一个 Product 类，该类实现了 Loggable 接口，给出了 logString()方法的具体实现，在日志中输出该商品的基本信息，代码如下所示：

```
<?php
class Product implements Loggable
{
    private $name;
    private $price;
    private $expiryDate;
    function __construct($name,$price,$expiryDate)
    {
        $this->name = $name;
        $this->price = $price;
        $this->expiryDate = $expiryDate;
    }
    function logString()
    {
        printf("商品名称:%1\$s  单价: %2\$s 元  生产日期: %3\$s  <br/>",$this
->name,$this->price,$this->expiryDate);
    }
}
?>
```

(4) 定义一个 MyLog()方法，在该方法中判断传入对象是否是 Loggable 接口的实现对象，是的话，调用输出日志信息的方法 logString()，否则的话，输出错误的提示信息。代码如下所示：

```
<?php
function MyLog($obj)
{
    if($obj instanceof Loggable)
    {
        $obj->logString();
    }else{
        print "对不起,您没有获取该服务许可证<br/>";
    }
```

```
}
?>
```

(5) 分别创建 Person 类和 Product 类对象，调用 MyLog()方法，将 Person 类对象和 Product 类对象作为参数传入 MyLog()方法中，记录对象的日志信息，代码如下所示：

```
<?php
    $person = new Person("小月","郑州市","01230",24);
    $product = new Product("夹心饼干",3.50,"2010-04-23");
    MyLog($person);
    MyLog($product);
?>
```

3.6.4　运行结果

打开 IE 浏览器，在地址栏中输入“http://localhost/PHP_3/interface/interface.php”，可以看到输出的用户和商品信息，如图 3-5 所示。



图 3-5　用户和商品信息

3.6.5　实例分析



源码解析：

本实例首先定义一个输出日志信息接口 Loggable，在该接口中定义输出日志信息方法 logString()，然后创建 Person 类和 Product 类，这两个类分别实现了 Loggable 接口，最后定义了一个 MyLog()方法，该方法接受 Loggable 类型对象，输出日志信息。

3.7　抽象出动物的特征

有时基类并不与具体的事物相联系，而是只表达一种抽象的概念，为其派生类提供一个公

共的界面，为此 PHP 中引入了抽象类的概念。抽象类不能实例化，只能作为由其他类继承的基类。本节将详细介绍抽象类的定义与应用。

 **视频教学：** 光盘\videos\03\抽象类.avi　　长度：5 分钟

3.7.1　基础知识——抽象类

抽象类中可以定义抽象方法，抽象方法与一般方法不同，抽象方法没有任何代码。如果类包含抽象方法，那么该类也必须声明是抽象类。在 PHP 中，抽象类和抽象方法都用 abstract 关键字声明。

下面创建一个示例，具体讲解如何创建抽象类和抽象方法，以及如何实现抽象类中的抽象方法。代码如下所示：

```
<?php
    abstract class Shape{
        abstract protected function get_area();     //计算面积方法
    }

    class Rectangle extends Shape{
        private $width; private $height;
        function __construct($width, $height)
        {
            $this->width=$width;
            $this->height=$height;
        }
        function get_area()                          //计算长方形面积
        {
            echo $this->width*$this->height;
        }
    }
    $Shape_Rect = new Rectangle(20,30);
    $Shape_Rect->get_area();
?>
```

3.7.2　实例描述

前段时间和一个朋友一起做了个公益网站，号召大家关爱动物，正好用到了 PHP 的抽象特性，封装了一个抽象的动物类，其他具体的动物类都继承这个抽象的动物类，这样代码看起来十分简洁，而且代码结构层次十分清晰，这样也使编程过程中出错几率大大降低。本节实例就来演示抽象的具体应用。

3.7.3　实例应用

【例 3-6】 抽象出动物的特征

(1) 新建一个 abstract.php 文件，在该文件中抽象出一个 Animal 类，定义动物名称变量 $name，并用一个抽象方法 enjoy()来展现动物的不同特征，代码如下所示：

```
abstract class Animal //因为这个类有抽象方法,所以这个类一定要声明为抽象类
{
    private $name;
    public function __construct($name)
    {
        $this->name = $name;
    }
    public abstract function enjoy(); //把这个方法定义为抽象方法,用来被重写
}
```

(2) 新建一个 Cat 类，该类继承了 Animal 类的$name 属性，还给出了 Cat 类特有的属性——猫的眼睛颜色$eyesColor，并重写了抽象方法 enjoy()，在该方法中展现了猫的特征，代码如下所示：

```
class Cat extends Animal //类里面的方法不想自己去实现
{
    //也可以把 Cat 定义为 abstract 类,把它的方法定义为抽象的
    private $eyesColor;//让继承它的类去实现
    public function __construct($name, $eyesColor)
    {
        $this->eyesColor = $eyesColor;
    }
    public function enjoy()
    {
        echo("猫的叫声,咪咪……");
    }
}
```

(3) 新建一个 Dog 类，该类也继承自 Animal 类，添加了一个 Dog 类特有的属性$furColor，重写了抽象方法 enjoy()，来展示狗的一些特征，代码如下所示：

```
class Dog extends Animal
{
    private $furColor;
    public function __construct($name, $furColor)
    {
        $this->furColor = $furColor;
    }
    public function enjoy()
    {
        echo("狗的叫声, 汪汪……<br />");
    }
}
```

(4) 新建一个 Bird 类，在该类中添加一个鸟特有的会飞属性$fly，重写了 enjoy()方法，来展示鸟的一些特征，代码如下所示：

```
class Bird extends Animal
{
    private $fly;
    public function __construct($name, $fly)
    {
        $this->fly = $fly;
    }
    public function enjoy()
    {
        echo("鸟飞得高，唧唧喳喳……<br />");
    }
}
```

(5) 分别创建动物类的实例，调用 enjoy()方法展示各自的特征，代码如下所示：

```
$cat = new Cat("小猫咪","黑眼睛");
$dog = new Dog("小狗狗","白毛");
$bird = new Bird("小鸟儿","会飞");
$cat->enjoy();
$dog->enjoy();
$bird->enjoy();
```

3.7.4 运行结果

打开 IE 浏览器，在地址栏中输入“http://localhost/PHP_3/abstract/abstract.php”，执行效果如图 3-6 所示。

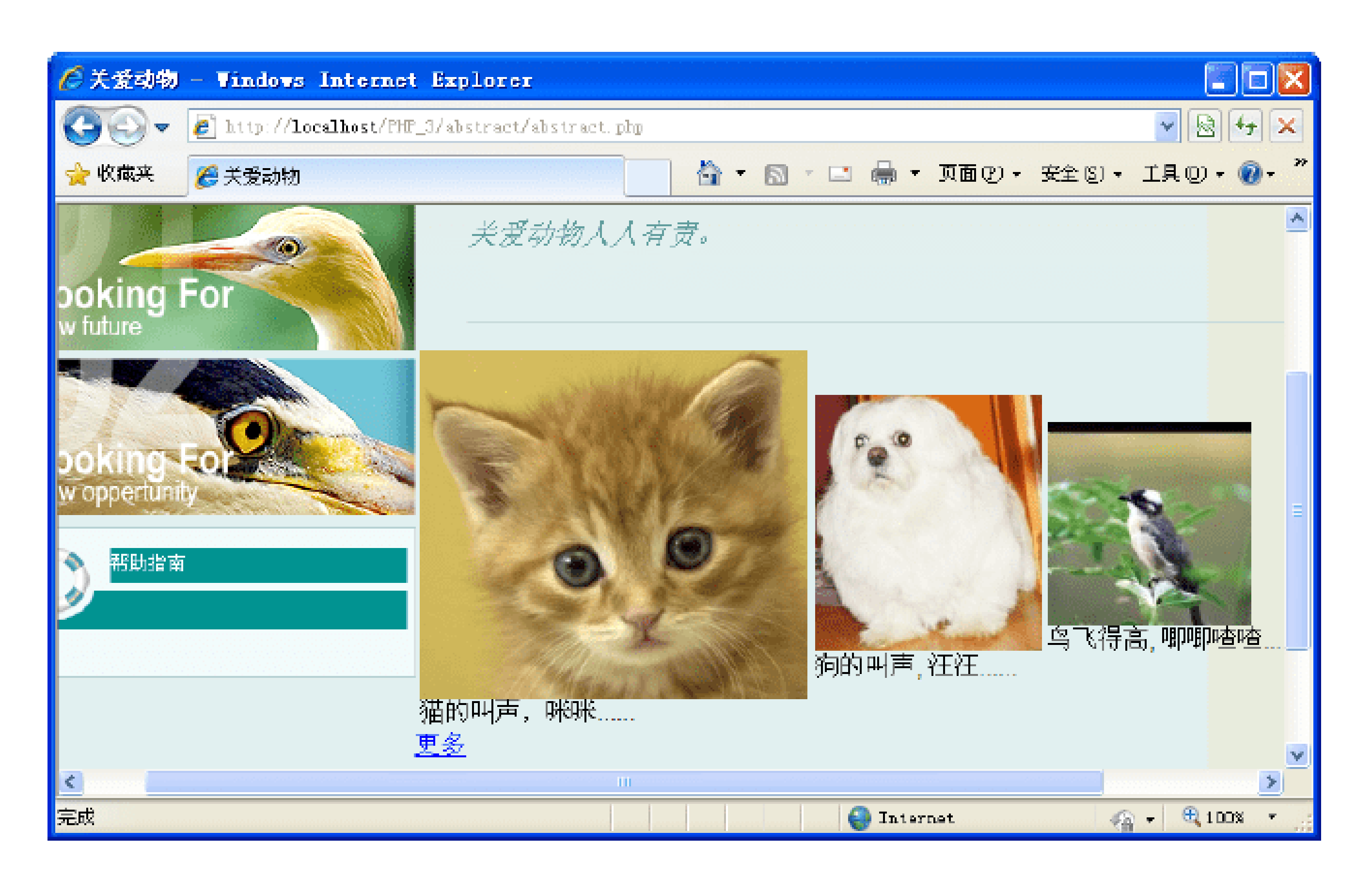
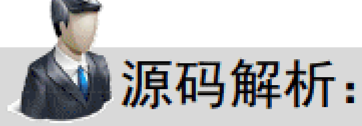


图 3-6 动物特征

3.7.5 实例分析


源码解析：

本实例首先定义了一个抽象 Animal 类，定义了一个$name 属性，一个带$name 参数的构

造函数，一个抽象方法 enjoy()。然后定义了一些具体的动物类，如 Cat、Dog 等。这些类继承了 Animal 类，分别添加了各自特有的属性，并重写 enjoy()方法展示各自的特征。

3.8　克 隆 文 件

PHP 4 面向对象功能最大的缺点之一是将对象视为另一种数据类型，这使得很多常见的 OOP 方法无法使用，如设计模式。幸运的是，PHP 5 解决了这个问题，提供了一种克隆(clone)对象的显示方法，也可以说是将一个对象的状态复制到另一个对象中。

 **视频教学：**光盘\videos\03\克隆方法.avi　　　　长度：9 分钟

3.8.1　基础知识——对象克隆

可以在对象前面加 clone 关键字来克隆对象，如：

```
destinationobject = clone targetobject;
```

下面通过一个面包类 bread，在该类中声明了$name(名字)和$taste(口味)字段，并分别给出这两个字段的设置和获取方法，代码如下所示：

```
<?php
    class Bread{
        private $name;
        private $taste;
        function setName($name)
        {
            $this->name = $name;
        }
        function getName()
        {
            return $this->name;
        }
        function setTaste($taste)
        {
            $this->taste = $taste;
        }
        function getTaste()
        {
            return $this->taste;
        }
    }
    $bread1 = new Bread();
    $bread1->setName("草莓奶油面包");
    $bread1->setTaste("甜甜的");
    $bread2 = clone $bread1;            //克隆面包对象
    $bread2->setName("牛肉松面包");
```

```
    $bread2->setTaste("咸咸的");

    echo $bread1->getName()." 口味: ".$bread1->getTaste()."<br/>";
    echo $bread2->getName()." 口味: ".$bread2->getTaste()."<br/>";
?>
```

在对象的克隆期间，有时不希望将所有的属性克隆过来，只是希望在某些局部位置做一点改动，如一个字段的值。这时可以在克隆的类中定义一个方法__clone()，从而调整对象的克隆行为。在将现有的对象成员复制到目标对象之后，会执行__clone()方法指定的操作。

下面通过一个示例，演示__clone()方法的使用，代码如下所示：

```
<?php
    class Bread{
        public static $bid = 1;

        function __construct()
        {
            $this->bid = self::$bid++;
        }
        function __clone(){
            $this->taste = "咸咸的";
            $this->bid = self::$bid++;
        }
    }
    $b1= new Bread();
    $b1->name="草莓奶油面包";
    $b1->taste="甜甜的";
    print $b1->bid."<br/>";
    print $b1->taste."<br/>";
    $b1_cloned = clone $b1;
    print $b1_cloned->bid."<br/>";
    print $b1_cloned->name."<br/>";
    print $b1_cloned->taste."<br/>";
?>
```

上述代码创建了一个 Bread 类，并实例化了一个$b1 对象，还设置了 name 和 taste。通过 clone 关键字克隆了一个$b1_cloned 对象，在克隆$b1_cloned 对象时，还执行了__clone()方法，将 taste 字段的值设置为“咸咸的”。

3.8.2 实例描述

很早以前就听说过克隆，上学的时候课本上也提到了克隆羊的出现，当时对克隆产生一种敬畏，心里想可以根据现有生物克隆出一个一模一样的生物出来，太不可思议了。现在编程中也支持克隆，可以根据现有的对象克隆出一个一模一样的对象出来。本节实例就来看看如何克隆出一个文件对象。

3.8.3　实例应用

【例 3-7】 克隆文件

(1) 新建一个 clone.php 文件，在该文件中定义了一个 MyFile 类，声明了一个$file_name 和$file_handle 字段，定义了一个设置文件名称函数 setFileName()，并定义了 openFileForReading()函数用来读取文件内容，__clone()函数用来克隆文件对象，具体代码如下所示：

```
<?php
class MyFile {
    private $file_name;
    private $file_handle = NULL;
    function setFileName($file_name)        //设置文件名称
    {
        $this->file_name = $file_name;
    }

    function openFileForReading()           //打开文件读取文件
    {
        $this->file_handle = fopen($this->file_name, "r");
        $lines = file($this->file_name);    //file()函数一行一行来读取文件数据，
                                              该函数将在后面章节具体讲解
        foreach($lines as $line)
        {
            echo $line."<br/>";             //输出文件内容
        }
    }

    function __clone()                      //克隆文件
    {
        if ($this->file_handle) {
            $this->file_handle = fopen($this->file_name, "r");
        }
    }
}
?>
```

(2) 创建一个 MyFile 文件对象$file1，并使用 clone 关键字克隆一个新文件$file2，重新设置$file2 的文件名字，并调用 openFileForReading()方法来读取文件，代码如下所示：

```
<?php
    $file1 = new MyFile();
    $file1->setFileName("file1.txt");
    $file2 = clone $file1;
    $file2->setFileName("file2.txt");
    $file2->openFileForReading();
?>
```

3.8.4 运行结果

打开 IE 浏览器，在地址栏中输入“http://localhost/PHP_3/clone/clone.php”，执行效果如图 3-7 所示。



图 3-7 克隆文件

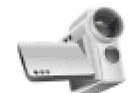
3.8.5 实例分析


源码解析：

本实例首先创建了一个 MyFile 类，在该类中定义了一个$file_name 和$file_handle 字段，然后为该类定义了设置文件名称函数 setFileName()，并定义了 openFileForReading()函数来读取文件内容，__clone()函数用来克隆文件对象。最后创建了一个文件对象$file1，通过$file2 = clone $file1 语句，克隆出一个新的文件对象$file2，并调用 setFileName()函数，重新设置文件名称，读取文件内容。

3.9 使用反射实现代理

反射是操纵面向对象范型中元模型的 API，其功能十分强大，可帮助我们构建复杂、可扩展的应用。其用途有自动加载插件、自动生成文档以及扩充 PHP 语言。


视频教学：光盘\videos\03\反射.avi　　长度：5 分钟

3.9.1　基础知识——反射

PHP 反射 API 由 4 个类组成，即 ReflectionClass、ReflectionMethod、ReflectionParameter 和 ReflectionProperty。可以帮助我们访问程序的元数据或者同相关的注释进行交互。借助反射，我们可以获取类实现了哪些方法，创建一个类的实例(不同于用 new 创建)，调用一个方法(也不同于常规调用)、传递参数等。本节将详细介绍这些反射 API 的使用。

1．ReflectionClass 类

ReflectionClass 类主要用于了解类的相关信息，例如获得该类的方法和成员的列表以及验证类是否为最终类(final)等。

ReflectionClass 类继承于 Reflector 接口，只需要实例化该类，就可以访问类中的成员，从而获取类的基本信息。下面定义一个 Teacher 类，声明了两个字段和 4 个方法，最后创建一个 ReflectionClass 实例，用来获取 Teacher 类的信息，具体代码如下所示：

```
<?php
    class Teacher {
        private $name;
        private $subject;
        public function getName(){
            return $this->name;
        }
        public function setName($name){
            $this->name = $name;
        }
        public function getSubject(){
            return $this->subject;
        }
        public function setSubject($subject){
            $this->subject = $subject;
        }
    }

    $class = new ReflectionClass('Teacher');
    $methods = $class->getMethods();
    echo "Teacher 类的方法有：<br/>";
    foreach($methods as $method)
        echo $method->getName()."<br/>";
    $isAbstract = $class->isAbstract()?"是一个抽象类":"不是一个抽象类";
    $isFinal = $class->isFinal()?"是一个最终类":"不是一个最终类";
    echo("<br/>");
    echo $class->getName()."类".$isAbstract."<br/>";
    echo $class->getName()."类".$isFinal;
?>
```

上述代码使用$class = new ReflectionClass('Teacher')语句创建了反射类的实例化对象$class，然后通过该对象调用 getMethods()方法获取类中方法，getName()方法获取方法的名称，

isAbstract()方法判断该类是否为抽象类，最后使用 isFinal()方法判断该类是否为最终类。

2. ReflectionMethod 类

ReflectionMethod 类主要用于获取方法的详细信息，例如该方法是否为抽象方法、最终方法、静态方法等。下面创建一个实例，演示获取一个方法的详细信息。具体代码如下所示：

```
<?php
    class Conditioner{
        private $style = '立式空调';
        public function cold(){
            echo "空调样式为：".$this->style.",它可以制冷！";
        }
    }

    $method = new ReflectionMethod('Conditioner','cold');
    $isPublic = $method->isPublic()?"是":"否";
    printf("方法%1\$s 是否为公共方法：%2\$s<br/>",$method->getName(),$isPublic);
    printf("方法的参数个数：%1\$s",$method->getNumberofParameters());
?>
```

上述代码创建了一个 Conditioner 类，声明了一个$style 字段和一个 cold()方法，然后通过$method = new ReflectionMethod('Conditioner','cold')语句，创建一个反射方法类对象$method，并通过该对象获取 cold()方法信息。

3. ReflectionParameter 类

ReflectionParameter 类主要用于获取一个函数或方法的参数信息，如参数的名称。下面通过一个示例演示 ReflectionParameter 类的使用，代码如下所示：

```
<?php
    class BigNum
    {
        public function compare($number)
        {
            if($number>100)
            {
                echo $number."大于 100<br/>";
            }
            else{
                echo $number."小于 100<br/>";
            }
        }
    }
    $method = new ReflectionMethod("BigNum","compare");
    $parameters = $method->getParameters();
    foreach($parameters as $parameter)
        echo $parameter->getName()."<br/>";
?>
```

上述代码创建一个 BigNum 类，定义了一个 compare()方法，然后实例化一个 ReflectionMethod 类对象$method，调用 getParamters()方法获得 compare()方法的所有参数的实例对象，并通过 foreach 循环依次取出每个参数对应的实例对象，调用 getName()方法获取参数的名称。

4．ReflectionProperty 类

ReflectionProperty 类主要用于了解特定类的属性，即获取类的属性的相关信息，例如获取属性的名称、数值和作用域范围等。

下面定义一个 Employee 类，并声明一个$name 属性，然后创建 ReflectionProperty 类的一个实例对象$prop，通过调用该对象的一些方法获取属性的信息，具体代码如下所示：

```
<?php
    class Employee
    {
        public $name;   //员工姓名
    }
    $prop = new ReflectionProperty('Employee','name');
    printf("属性作用域: %1\$s %2\$s %3\$s %4\$s <br/>名称: %5\$s",
            $prop->isPublic()?' public':'',
            $prop->isPrivate()?' private':'',
            $prop->isProtected()?' protected':'',
            $prop->isStatic()?' static':'',
            $prop->getName());
?>
```

5．ReflectionExtension 类

ReflectionExtension 类允许反向设计扩展。你可以在使用 get_loaded_extensions()运行的时候重新返回所有加载的扩展。为了内省一个扩展，你必须首先创建 ReflectionExtension 类的一个实例。下面创建一个该类的实例来访问类成员，代码如下所示：

```
<?php
    $extension = new ReflectionExtension('date');
    printf("名称: %s<br/>".
        "函数: [%d] %s<br/>".
        "类名称: [%d] %s<br/>",
        $extension->getName(),
        sizeof($extension->getFunctions()),
        var_export($extension->getFunctions(),1),       //取得该扩展的所有函数
        sizeof($extension->getClassNames()),
        var_export($extension->getClassNames(),1));     //得到该扩展的类名
?>
```

3.9.2　实例描述

现在“代理”这个词越来越风靡了，比如说层出不穷的各色产品代理商，PHP 也不忘记赶时髦，它提供的反射 API 也可以实现代理的功能，本节实例就演示了如何使用反射来实现代理。

3.9.3 实例应用

【例 3-8】 用反射实现代理

(1) 新建一个 reflection.php 文件，在该文件中定义一个 Agency 类，并为该类定义一个 callAgency()方法，代码如下所示：

```
class Agency {
    function callAgency() {
        print "竹子美食形象代理";
    }
}
```

(2) 在 reflection.php 文件中，再创建一个 AgencyDelegator 类，该类用来代替 Agency 类来实现它的 callAgency()方法，__call()方法会在代理对象调用执行相应方法时自动调用执行。具体代码实现如下所示：

```
class AgencyDelegator {
    private $targets;
    function addObject($obj) {
        $this->target[] = $obj;
    }
    function __call($name, $args) {
        foreach ($this->target as $obj) {
            $r = new ReflectionClass($obj);
            if ($method = $r->getMethod($name)) {
                if ($method->isPublic() && !$method->isAbstract()) {
                    return $method->invoke($obj, $args);
                }
            }
        }
    }
}
```

(3) 创建代理类 AgencyDelegator 的对象$obj，通过该对象调用 addObject()方法来添加需要代理执行的 Agency 类对象，然后代理执行 callAgency()方法，代码如下所示：

```
$obj = new AgencyDelegator();
$obj->addObject(new Agency());
$obj->callAgency();
```

3.9.4 运行结果

打开 IE 浏览器，在地址栏中输入“http://localhost/PHP_3/reflection/reflection.php”，可以看到在页面中央输出反射代理方法的内容，执行效果如图 3-8 所示。



图 3-8 反射实现代理

3.9.5 实例分析

**源码解析：**

本实例首先创建了一个 Agency 类，在该类中定义了一个 callAgency()方法，然后创建了一个代理类 AgencyDelegator，在该类中定义一个 addObject()方法，用来添加需要代理的类对象，还定义了一个__call()方法，用来执行需要代理的类的方法。最后创建一个 AgencyDelegator 类对象$obj，调用 addObject()方法添加 Agency 类对象，当代理类对象$obj 调用 Agency 类的 callAgency()方法时，自动执行__call()方法。

3.10 类/对象函数

通过上面的学习，PHP 的面向对象就学得差不多了，本章最后再来看看 PHP 5 提供的一些内置的用来操作类和对象的函数。

**视频教学：**光盘\videos\03\类和对象函数.avi 长度：8 分钟

使用 PHP 开发大型项目，实体类或对象会比较多，不方便管理，在 PHP 5 中，提供了一些内置函数，解决了这个问题，例如 class_exits()、get_class()、is_subclass_of()和 method_exists()等函数，本节将介绍主要内置函数的使用。

1. class_exits()函数

class_exits 函数用于检查类是否已定义，如果由 class_name 所指的类已经定义，则此函数

返回 true，否则返回 false。该函数的语法格式如下所示：

```
boolean class_exists(string $class_name[,boolean $autoload])
```

2. get_class()函数

get_class()函数用于返回对象实例 object 所属类的名字。如果 object 不是一个对象则返回 false。该函数的语法格式如下所示：

```
string get_class(object object)
```

3. get_class_methods()函数

get_class_methods()函数返回由 class_name 指定的类中定义的方法名所组成的数组。如果执行出现错误，则返回 null。该函数语法格式如下所示：

```
array get_class_methods (mixed $class_name)
```

4. get_class_vars()函数

get_class_vars()函数返回由类的默认公有属性组成的关联数组，此数组的元素以 varname => value 的形式存在。该函数的语法格式如下所示：

```
array get_class_vars  (string $class_name)
```

5. get_declared_classes()函数

get_declared_classes()函数返回由当前脚本中已定义类的名字组成的数组。根据 PHP 发行包的配置不同，该函数输出会有所区别。该函数的语法格式如下所示：

```
array get_declared_classes(void)
```

6. get_object_vars()函数

get_object_vars()函数返回由 object 指定的对象中定义的属性组成的关联数组。在 PHP 4.2.0 之前的版本中，如果在 object 对象实例中声明的变量没有被赋值，则它们将不会在返回的数组中。而在 PHP 4.2.0 之后，这些变量作为键名将被赋予 null 值。该函数的语法格式如下所示：

```
array get_object_vars(object object)
```

7. method_exists()函数

method_exists()函数用于检查类的方法是否存在。如果 method_name 所指的方法在 object 所指的对象类中已定义，则返回 true，否则返回 false。该函数的语法格式如下所示：

```
boolean method_exists(object object,string method_name)
```

8. interface_exists()函数

interface_exists()函数用于检查接口是否已被定义，本函数在由 interface_name 给出的接口已定义时返回 true，否则返回 false。该函数的语法格式如下所示：

```
boolean interface_exists(string interface_name[,boolean autoload])
```

3.11　常见问题解答

3.11.1　PHP extends 继承多个类怎么写



PHP extends 继承多个类怎么写？
网络课堂：http://bbs.itzcn.com/thread-12422-1-1.html

```
class A extends B{
}
```

若想让 A 继承 B 和 C，使用 PHP 该怎么编写？

【解决办法】：PHP 和 Java 一样，只能单继承，想实现多继承只能通过多个接口来实现。

3.11.2　PHP 子类的构造函数能继承父类的吗



PHP 子类的构造函数能继承父类的吗?
网络课堂：http://bbs.itzcn.com/thread-12424-1-1.html

本人初学 PHP，想问一下 PHP 子类的构造函数能继承父类的吗?

【解决办法】：可以继承的，你加上 parent::_construct()，就可以访问继承过来的父类构造函数了。

3.11.3　PHP 类外函数调用类的成员函数的方法



PHP 类外函数调用类的成员函数的方法?
网络课堂：http://bbs.itzcn.com/thread-12425-1-1.html

有一个数据库操作类 class mySql_Class，并且已经实例化，类外函数怎样调用类的成员函数(前提是不在类外函数内再次实例化类)?

【解决办法】：这个并不难，只需使用一个 global 关键字，代码如下所示：

```
$c=new mySql_Class;
function x()
{
    global $c;
    $c->doMethod();
}
```

3.12　习　　题

一、填空题

(1)　下面代码创建了一个圆类 Circum，定义了一个 girth()方法来计算圆的周长，在该方法

中需要用到一个常量 PI，请在类中声明一个常量 PI。

```
class Circum
{
    __________ PI = '3.1415926';
    public function girth($r)
    {
        return 2*self::PI*$r;       //计算圆的周长
    }
}
```

(2) 在 PHP 中可以使用 echo 和____________在页面上输出一些信息。

(3) 下面代码的输出结果为____________。

```
<?php
    class User
    {
        private $username;
        public function getUsername()
        {
            return $this->username;
        }
        public function setUsername()
        {
            $this->username = $username;
        }
    }
    $user = new User();
    $user->setUsername("hello");
    echo ($user->getUsername());
?>
```

二、选择题

(1) 将下面定义一个可用的 User 接口的代码补充完整____________。

```
____________ User{
    public function getName();
    public function setName($_name);
    public function getDiscount();
}
```

A．extends　　B．const　　C．interface　　D．abstract

(2) 下面的关键字可以用来克隆对象的是____________。

A．clone　　B．construct　　C．extends　　D．abstract

(3) 下面函数可以销毁对象的是____________。

A．function __construct($name){
        $this -> object_name = $name;
    }

B．function __clone(){

```
        $this->taste = "咸咸的";
        $this->bid = self::$bid++;
    }
C．static function getCount(){
        return self:$count;
    }
D．function __destuct(){
        print("释放内存中的所有对象");
    }
```

三、上机练习

上机练习： 显卡接口的定义与应用。

练习要求：首先定义显卡的功能接口，该接口包含两个方法声明 Display()和 getName()，然后定义一个显卡的生产厂家类来实现显卡接口。最后定义一个主板类，主板类只需要用这些接口就行了，并为主板类定义一个运行方法 run()，该方法接收一个 VideoCardInter 接口类型的参数，运行主板。执行效果如图 3-9 所示。

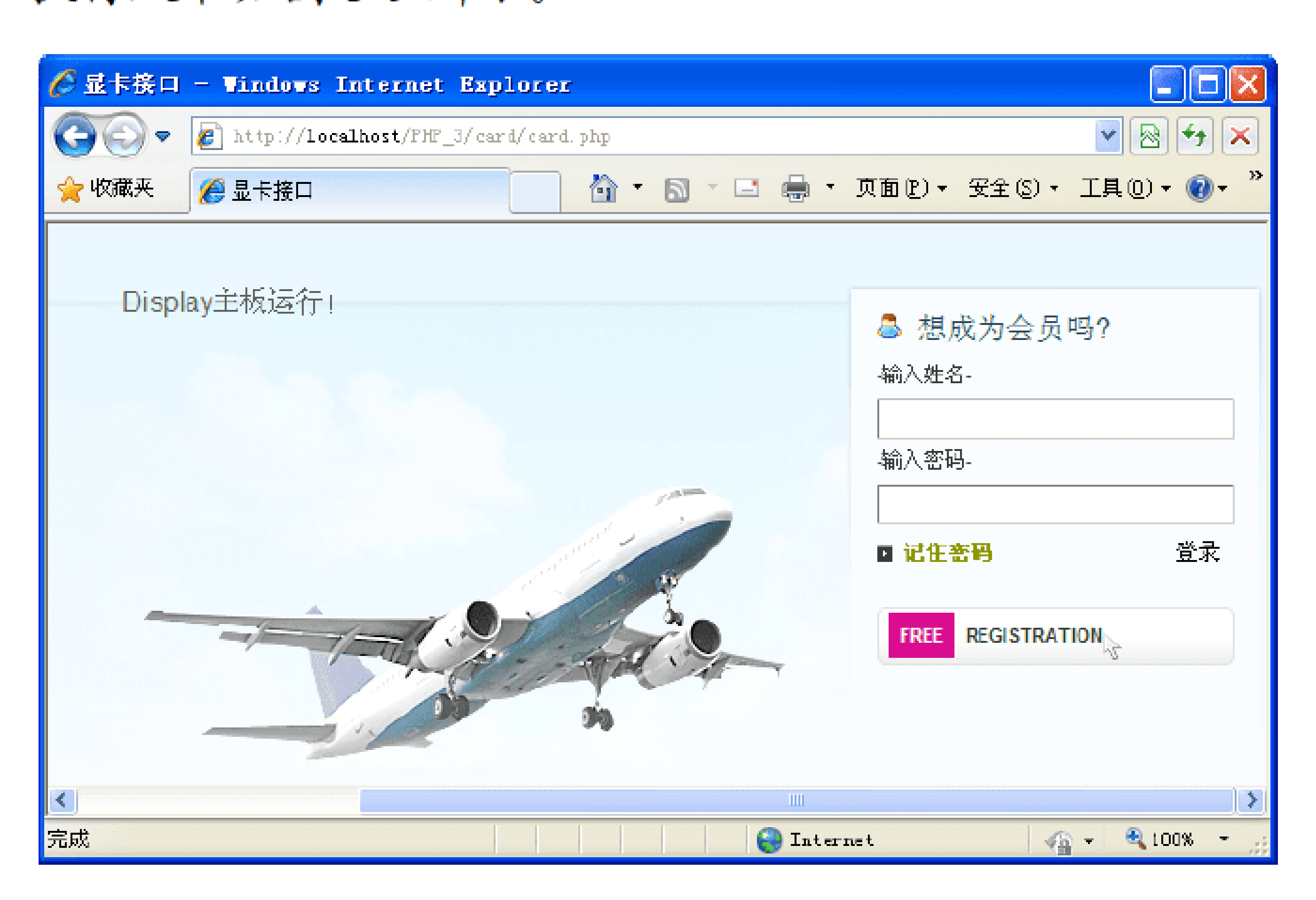
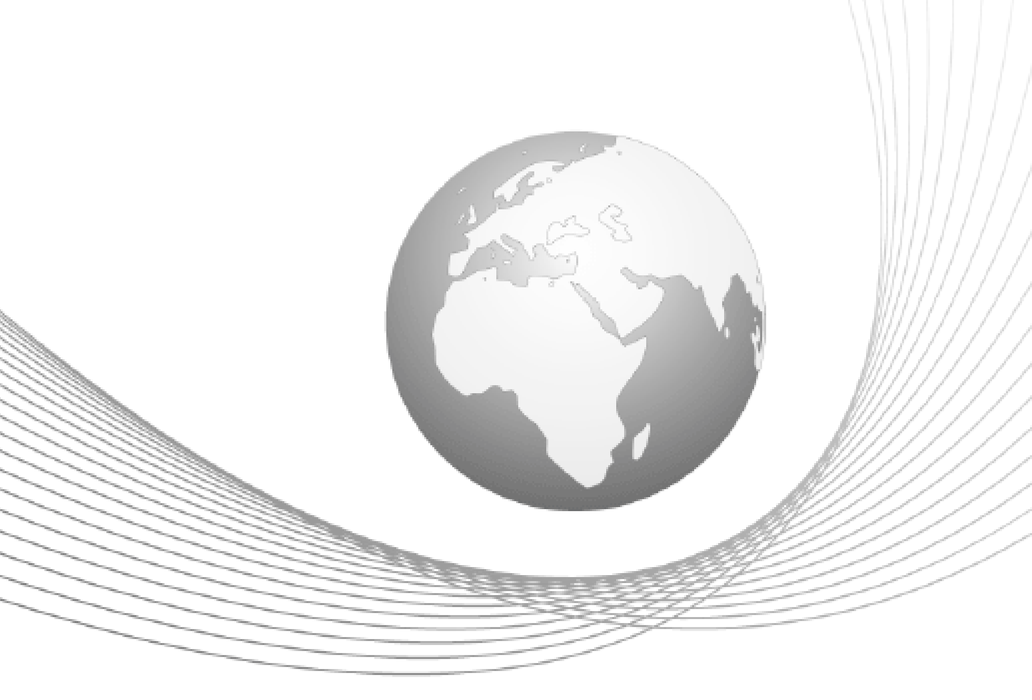


图 3-9　运行主板



第 4 章　错误和异常处理

内容摘要：

在程序的开发中，难免会因某种原因而产生错误。有些错误是程序员引起的，是在开发过程中失误的产物；还有一些是用户引起的，是由用户终端不愿意或无法遵循应用程序约束而造成的。因此，避免、调试、修复错误和对程序可能发生的异常进行处理是程序员的一种必备能力。PHP 中提供了良好的错误提示，本章将对 PHP 的错误配置指令、错误日志和异常处理进行详细的讲解。

学习目标：

- 掌握 PHP 中与错误有关的配置指令。
- 掌握 PHP 中错误日志的分析。
- 掌握 PHP 中常见的错误异常处理。

4.1 配 置 指 令

PHP 为报告和记录在编译和运行时生成的错误、警告和提示提供了一种方便灵活的方法。开发人员可以控制报告的敏感度，例如这些信息是否在浏览器中显示及如何显示，信息将记录在文件中还是系统日志中等。这些都需要用到 PHP 中与错误有关的配置指令，我们在这一节中进行介绍。

**视频教学：**光盘\videos\04\配置指令.avi　　　　长度：8 分钟

在 PHP 中，与错误有关的配置指令确定了 PHP 的错误报告行为。这些指令有许多，我们在这里介绍其中的一些。

1. error_reporting(string)

error_reporting 指令是确定错误报告的敏感度级别，它的作用域是 PHP_INI_ALL，默认值为 E_ALL& ~E_NOTICE & ~E_STRICT。这种级别共有 13 种，如表 4-1 所示。这些级别的任意组合在 PHP 的错误报告中都有效，而且每一种级别都包括位于其下的所有级别。

表 4-1　PHP 的错误报告级别

级 别 名	级别描述
E_ALL	报告所有错误和报告
E_ERROR	报告致命的运行时错误
E_WARNIG	报告非致命的运行时错误
E_PARSE	报告编译时解析错误
E_MOTICE	报告运行时注意信息，如未初始化变量
E_STRICT	PHP 版本可移植性建议
E_CORE_ERROR	报告在 PHP 启动时发生的致命错误
E_CORE_WARNING	报告在 PHP 启动时发生的非致命性错误
E_COMPILE_ERROR	报告致命的编译时错误
E_COMPILE_WARNING	报告非致命的编译时错误
E_USER_ERROR	报告用户引发的致命错误信息
E_USER_WARNING	报告用户引发的非致命错误信息
E_USER_NOTICE	报告用户引发的注意消息

假设我们只考虑致命的运行时错误、解析错误和核心错误，就可以使用逻辑操作符将指令设置为：

```
error_reporting E_ ERROR | E_ PARSE | E_ CORE_ERROR
```

2. display_errors(on,off)

display_errors 指令是输出 error_reporting 指定级别上的所有错误，作用域是 PHP_INI_ALL，

默认为 on。这个指令只能在程序的测试阶段使用，当 Web 应用投入使用时，禁止使用此指令。例如用文本文件存储用户的电话，应用程序就无法写入文件，而 PHP 就会向用户终端报告这个问题，并输出如下的错误信息：

```
Warning:fopen(telphone.txt):failed to open stream: Permission denied in /home/
www/htdocs/pmnp/40/displayerrors.php on line 45
```

这样一来，用户就可以知道这个文档的确切位置，就可以通过输入 http://www.example.com/telphone.txt 的地址来得到文件中的所用信息了。

3．log_errors(on,off)

log_errors(on,off)指令是告诉 PHP 应当记录的错误信息，并保存到某个文件或 syslog 中。这些记录能为确定应用程序和 PHP 引擎特定的问题提供最有价值的信息，日志语句记录的位置取决于 error_log 指令。作用域是 PHP_INI_ALL，默认为 off。

4．error_log(string)

error_log 指令启用时，将指定消息目标，PHP 支持将错误记录在指定文件或系统日志中，在 Windows 下，将本参数设置为 syslog 会把消息记录在事件日志中。这些日志可以通过事件查看器来查看。作用域是 PHP_INI_ALL，默认为 null。

5．display_startup_errors(on,off)

display_startup_errors 指令会显示 PHP 引擎初始化时遇到的所有错误，禁用时可防止将启动过程中的错误显示给用户，所以此指令只能在测试时使用。作用域是 PHP_INI_ALL，默认为 off。

6．log_errors_max_len(integer)

log_errors_max_len(integer)指令用于设置每个日志项的最大长度，以字节为单位。参数设置为 0 时则不限制最大长度。作用域是 PHP_INI_ALL，默认为 1024。

7．ignore_repeated_errors(on,off)

ignore_repeated_errors 指令用于在程序运行时，防止 PHP 记录在同一文件中同一行上发生的重复错误信息。作用域是 PHP_INI_ALL，默认为 off。

8．ignore_repeated_source(on,off)

ignore_repeated_source 指令用于 PHP 在忽略重复错误时将不考虑错误的来源，而且每一个错误只能记录一次。作用域是 PHP_INI_ALL，默认为 off。

9．track_errors(on,off)

track_errors(on,off)指令的作用会让 PHP 在$php_error_msg 变量中存储最近的错误信息，其作用域仅限于出现错误的特定脚本内，一旦注册，就可以随心所欲地使用此变量数据。例如存储到数据库或其他操作。作用域是 PHP_INI_ALL，默认为 off。

4.2　代码中常见错误分析

我们在编写程序时，程序会由于程序员的能力出现这样或那样的错误，这些错误通常会迷惑 PHP 编译器。虽然 PHP 会对错误报告出错误信息，但是很多程序员由于并不了解这些错误信息所表示的含义，而对错误无从下手。在这一节里我们将对 PHP 编程过程中常见的错误信息进行详细的分析与解决，希望对开发者有所帮助。


视频教学：光盘\videos\04\常见错误分析.avi　　　　长度：6 分钟

4.2.1　基础知识——常见错误分析

PHP 的程序开发中，经常出现的错误大致可以分为以下几类：语法错误、语义错误、逻辑错误、注释错误和运行时错误。其中前四项对于资深的程序员来说很好解决。下面我们来看看实际应用中的几个常见错误。

1．变量声明错误

变量声明错误属于语法错误，是在程序中使用了错误的语法而导致的错误。如果在一条语句中声明一个变量的具体方式如下：

```
$str = "1987年";
```

则程序在运行时，编译器首先会求出语句右半部分的值，如果这部分的语法不正确，常常会出现对变量的解析错误。例如，如下是一个变量声明错误产生的信息。

```
Parse error:解析错误，unexpected T_WHILE in c:\\program files\\apache group\\
apache\\www\\4.2.php on line 3
```

每次确定了前一错误时，解析错误一个接一个地不断出现。而且，解析错误具有很少的信息，几乎不报告错误所在的行号。

2．使用预定义字词产生的错误

PHP 的预定义词有很多，如 if、while、function 等。如果要使用预定义词作为变量的话，PHP 解析器在解析时会报告出更多的错误。例如下面的代码：

```
<?php
  $sd = djdjdk
 while ($sd=djdjdk){
    echo "chdkddk";
 }
?>
```

当出现错误时，我们会直接看到错误可能出现在第 2 行，但编译器提示的信息则会和我们想到的有偏差，编译器提示的错误信息如下所示：

```
Parse error: parse error, unexpected T_WHILE in c:\\program files\\apache
```

```
group\\apache\\www\\4.2.php on line 3
```

在第 3 行中，很显然 while 语句的语法是正确的，那么编译器为什么会在这一行提示错误呢。那是因为编译器发现 PHP 预定义字 while 不应该在这一行出现。

3．程序语句结束错误

这种错误常常会出现在函数和循环控制语句中，是由于程序语句中缺少大括号“}”所引起的。当程序执行时，PHP 编译器会不断地查找表示结束的大括号直至到达文件末尾为止。因为编译器未找到一个匹配的大括号，就会报告文件末尾处有错误。例如下面的代码：

```
function user(){
    for($i=0;$i<100;$i++){
 }
?>
```

这段代码在编译器解析执行时，会提示如下的错误：

```
Parser error "'}' expected in compound-statement." test    4.2.php line 10
```

我们根据错误信息可以知道，在第 10 行的复合语句中出现“}”的结束语句错误。

4．MySQL 错误

我们在使用 PHP 进行与 MySQL 数据库的操作时，也会出现很多的错误，如数据查询失败、数据库连接失败等问题。引起这些错误的原因有很多，例如 sql 语句的书写错误或者是在连接数据库时输入了错误的用户名和密码等。下面我们来看一个例子。

```
while($row = mysql_fetch_array($result)) {
```

这段代码在解析器解析时，可能会输出以下的错误信息。

```
Warning: Supplied argument is not a valid MySQL result resource in...
```

这段错误信息告诉我们参数$result 不是一个有效的资源，这里我们就需要检查执行查询的 sql 语句是否有效才能解决问题。

4.2.2　实例描述

PHP 在实际的应用开发中，和 MySQL 数据库结合使用是最为普遍的，因此在这种情况下出现的错误也是很多的，下面我们通过一个实例来了解 PHP 与 MySQL 数据库使用中常见的错误和解决办法。

4.2.3　实例应用

【例 4-1】 解析在 PHP 中使用 MySQL 出现的错误

(1) 新建一个名为 insert.php 的页面，在页面中添加链接数据库的代码，如下所示：

```
session_start();
$conn=mysqli_connect("localhost","root","root","userman");
```

```
$conn->query('set names gb2312');
if(empty($conn))
{
   die("数据库连接失败");
}
$name=$_POST[username];
$pass=$_POST[userpass];
$pass2=$_POST[userpass2];
$email=$_POST[email];
$age = $_POST[age];
$str="select * from users where username='".$name."'";
$result1=$conn->query($str);
$row=$result1->fetch_row();
if($row)
{
    $temp="已有人注册此名,请重新选择名字!";
    echo $temp;
    echo"<a href=Register1.html>返回</a>";
}else{
     echo "注册成功";
}
```

(2) 新建一个 Register1.html 文件，在里面添加一个注册信息的表单，其部分代码如下：

```
<form action="./insert.php" method="post" id="apForm">
     <input type="hidden" name="referer" value="./index.html" />
     <div class="content">
       <p>
         <label for="email"> 您的登录名: </label>
         <input class="input" type="text" id=username maxLength=32 onBlur=
"checkusername();" name="username">
         <span id="checkusername"> </span> </p>
       <p>
         <label for="email"> 登录密码: </label>
         <input class="input" style="width:148px;" onkeyup=EvalPwdStrength
(this.value); onBlur="checkpassword()" type=password maxLength=16 id="password"
name="userpass">
         <span id="checkpassword"> </span> </p>

       <p>
         <label for="email"> 密码确认: </label>
         <input class="input" style="width:148px;" onBlur="checkpassword2()"
type=password maxLength=16 id="password2" name="userpass2">
         <span id="checkpassword2"> </span> </p>
       <p>
         <label for="email"> 电子邮箱: </label>
         <input id=email maxLength=32 onBlur="checkemail()" class="input"
name=email>
         <span id="checkemail"> </span> </p>
       <p>
         <label for="age"> 您的年龄: </label>
```

```
        <input class="input" type="text" id=age maxLength=32 name="age">
        </p>
      <p>
        <input type="hidden" name="registersubmit" value="true">
        <input type="submit" name="mysubmit" value="" class="tj">
      </p>
    </div>
  </form>
```

(3)　在 insert.php 页面中设置一个错误的数据库用户名和密码，并保存页面。

4.2.4　运行结果

将注册页面在浏览器中运行，填写好信息提交注册后，当在用于连接数据库的语句中输入错误的用户名和密码时，PHP 解析器会输出以下的错误信息。

```
Warning: mysql_pconnect() [function.mysql-pconnect]: Access denied for user
'root'@'localhost' (using password: YES) in D:\www\test/insert.php on line 3
```

这条错误信息告诉我们，在 insert.php 页面中的第 3 行出现链接数据库的用户名和密码错误，当我们找到这一行并进行正确的修改后，程序就可以正确地运行了。其结果如图 4-1 所示。



图 4-1　注册页面运行结果

当单击【注册】按钮后，程序会跳转到 inset.php 页面进行业务处理，PHP 解析器会正确地解析用户名和密码，并成功地连接数据库进行有效验证。

4.2.5　实例分析



源码解析：

上述实例中，我们讲解到了经常遇到的数据库连接错误 Warning: mysql_pconnect()及错误的处理方法。当然，在与数据库打交道时，出现的错误不只是这些，还需要我们进一步认真地研究。

4.3 处理表单输入异常

在程序开发、测试、运行及维护的各个阶段中，随时都可能出现不可预见的事件，从而打断正常的事件链，这些不可预见的事件称为异常。为了能正确、及时地处理出现的异常，许多编程语言通过抛出异常，也就是使用一个代码段来处理错误，然后对异常妥善地做出响应。而这种异常处理功能到 PHP 5.0 才被集成到该语言中。本节将介绍这一特性。

 **视频教学：**光盘\videos\04\php 异常处理.avi　　　　长度：6 分钟

4.3.1 基础知识——PHP 异常处理和异常类

在这一小节里，我们将讲到如何处理 PHP 的异常以及 PHP 的异常类的特性。

1. 如何处理异常

在很多的编程语言中，为了能够及时准确地处理异常，大多都会选择首先抛出这个异常，然后使用一段代码来处理这个错误。关于异常处理的过程，大多数编程语言将其抽象为 4 个步骤，具体如下所示。

(1) 应用程序尝试进行处理。

(2) 如果尝试失败，则将该异常抛出。

(3) 指定的异常处理器捕获异常，并进行相应的处理。

(4) 最后消除在尝试处理异常期间占用的资源。

虽然异常处理基本可以归纳为以上 4 个步骤，但在实际的程序开发中往往出现的异常并非只有一个，所以在很多时候可以创建多个异常处理器来解决这些问题。到目前为止，在 PHP 中只提供一个简单的 exception 处理器，根据需要可以扩展这个处理器。所以用户可以使用预定义处理器，或扩展某个预定义处理器来创建满足自己需要的定制处理器，代码如下所示：

```
try{
    if($row)
{
    $temp="已有人注册此名,请重新选择名字!";
    echo $temp;
    echo"<a href=Register1.html>返回</a>";
}else{
    echo "注册成功";
}
}catch(Execption $e){
    echo "代码出现异常";
}catch(emailExecption $me){
    echo "电子邮件格式不正确";
}
```

2. 基本异常类

PHP 的基本异常类的结构很简单，主要包括一个不带参数的默认构造函数、一个带有两个可选参数的重载构造函数和六个用于具体处理异常的方法。

1) 默认构造函数

基本异常类默认的构造函数不带参数，用户可以用如下方式使用异常类：

```
throw new Exception();
```

以上所抛出的异常在实例化后就可以使用异常类所具有的所有方法。

2) 重载构造函数

基本异常类的重载构造函数可以有两个参数，由此能提供默认构造函数所没有的其他功能。它的使用格式如下所示：

```
throw new Exception(message, error code);
```

这两个参数的功能如表 4-2 所示。

表 4-2　参数功能

参　数	功　能
message	作为一个对用户友好的解释，可以通过 getMessage()方法异常信息传递给用户，向用户提供友好的异常信息
error code	参数 error code 用于保存错误标识符，它可以映射到某个标识符表。而错误标识通常用于国际化和本地化。在这里错误标识符可以通过异常类的 getCode()方法得到

3. 基本异常类的六个方法

基本异常类中有六个用于具体处理异常的方法，这六个方法的具体作用如表 4-3 所示。

表 4-3　基本异常类的方法

方　法	作　用
getMessage()	用于返回传递给构造函数的错误消息
getCode()	用于返回传递给构造函数的错误代码
getLine()	用于返回出现错误时抛出异常的行号
getFile()	用于返回出现错误时抛出异常的文件名
getTrace()	用于返回一个数组，其中包括出现错误的上下文的有关信息。通常情况下，该数组包括文件名、行号、函数名和函数参数
getTraceAsString	用于返回与 getTrace()完全相同的信息，只是返回的信息是一个字符串而不是数组



因为基本异常类的声明都是 final，所以虽然可以扩展任何的异常类，但是不能重载它的任何一个方法。

4. 自定义扩展内置异常类

用户可以用自定义的异常处理类来扩展 PHP 内置的异常处理类。如果使用自定义的类来扩展内置异常处理类，并且要重新定义构造函数的话，建议同时调用 parent::construct()来检查所有的变量是否已被赋值。当对象要输出字符串的时候，可以重载_ _toString()并自定义输出的样式，代码如下所示：

```
class MyException extends Exception
{
    // 重定义构造器使 message 变为必须被指定的属性
    public function __construct($message, $code = 0) {
    // 确保所有变量都被正确赋值
       parent::__construct($message, $code);
    }
    // 自定义字符串输出的样式
    public function __toString() {
        return __CLASS__ . ": [{$this->code}]: {$this->message}\n";
    }
    public function customFunction() {
        echo "A Custom function for this type of exception\n";
    }
}
```

在这段代码中，我们自己定义了一个类 MyException，并继承自 Exception 类。

4.3.2 实例描述

在 PHP 的开发中，好的程序员会考虑到所有可能想到的情况，例如我们在处理从网站提供的 HTML 表单时，用户可以在表单中输入如用户名、密码和电子邮件等信息。例如输入电子邮件，用户可能会做如表 4-4 所示的事情。

表 4-4 用户动作

用 户	动 作
用户 A	输入了有效的电子邮件地址
用户 B	输入了不正确或无效的电子邮件地址
用户 C	没有输入任何电子邮件地址
用户 D	通过电子邮件攻击网站，如 SQL 注入等

下面就让我们来做一个处理注册表单的实例，看看通过 PHP 的异常处理机制怎样捕获多个异常。

4.3.3 实例应用

【例 4-2】 处理表单输入异常

(1) 新建一个名为 4.3Reg.html 的 HTML 文件，在文件中为页面设计一个表单，主要用来供用户输入用户名、密码、密码确认、电子邮箱和年龄。表单的部分代码如下：

```
<form action="register1.php" method="post" id="apForm">
     <input type="hidden" name="referer" value="./index.html" />
     <div class="content">
       <p>
         <label for="email"> 您的登录名: </label>
         <input  class="input"  type="text"  id=username  maxLength=32  onBlur=
"checkusername();" name="username">
         <span id="checkusername"> </span> </p>
       <p>
         <label for="email"> 登录密码: </label>
         <input  class="input"  style="width:148px;"  onkeyup=EvalPwdStrength
(this.value); onBlur="checkpassword()" type=password maxLength=16 id="password"
name="userpass">
         <span id="checkpassword"> </span> </p>

       <p>
         <label for="email"> 密码确认: </label>
         <input class="input" style="width:148px;"  onBlur="checkpassword2()"
type=password maxLength=16 id="password2" name="userpass2">
         <span id="checkpassword2"> </span> </p>
       <p>
         <label for="email"> 电子邮箱: </label>
         <input  id=email  maxLength=32   onBlur="checkemail()"  class="input"
name=email>
         <span id="checkemail"> </span> </p>
       <p>
         <label for="age"> 您的年龄: </label>
         <input class="input" type="text" id=age maxLength=32 name="age">
         </p>
         <p>
         <label for="yzm"> 输入验证码: </label>
<input class="input" type="text" id=authinput maxLength=15 style="width:
       50px" name="authinput">
       <img src=authimg.php?authnum=<? echo $authnum; ?>><br>
       </p>
```

(2) 新建一个名为 4.3regExp.php 的文件，在文件中，添加连接数据库的代码，如下所示：

```
session_start();
$conn=mysqli_connect("localhost","root","root","userman");
$conn->query('set names gb2312');
if(empty($conn))
{
   die("数据库连接失败");
}
```

(3) 获得表单中输入的数据，并能通过数据库进行查询，代码如下：

```
$name=$_POST['username'];
$pass=$_POST['userpass'];
$pass2=$_POST['userpass2'];
$email=$_POST['email'];
$age = $_POST['age'];
$str="select * from users where username='".$name."'";
$result1=$conn->query($str);
$row=$result1->fetch_row();
```

(4) 新声明一个处理邮件的自定义异常类 EmailException，在该类中定义处理邮件的方法，代码如下所示：

```
class EmailException extends Exception{
        function _construct($mess){
         $this->mess = $mess;
        }
       function emailMessage(){
        echo "电子邮件格式不正确";
       }
}
```

(5) 进行表单验证并捕获多个异常，其实现代码如下所示：

```
try{
        if($row){
          $temp="已有人注册此名,请重新选择名字!";
          echo $temp;
          echo"<a href=Register1.html>返回</a>";
          }else{
         echo "注册成功";
          }
          if($email=="pingniangan@yahoo.cn"){
          throw new EmailException();
       }
   }catch(Execption $e){
         echo "代码出现异常";
        echo $e->getMessage();
   }catch(EmailException $me){
       echo $me->getMessage();
   }
```

(6) 保存页面，这样我们就对用户输入邮件时所有可能出现的异常进行了处理。

4.3.4 运行结果

将表单页面在浏览器中运行，我们可以看到如图 4-2 所示的结果。

单击【注册】按钮后，如果用户输入的电子邮箱不正确或者没有输入，程序会将异常处理信息抛给浏览器，并显示给用户。

代码出现异常
电子邮件格式不正确



图 4-2　表单运行结果

4.3.5　实例分析

源码解析：

上述实例中，我们使用 PHP 的异常处理机制处理了用户输入邮件时所出现的异常。从程序中看出，程序完全可以出现两个不同的异常，一个是基本异常 Exeption，而另一个是自定义的异常 EmailException。

4.4　常见问题解答

4.4.1　怎样处理 unlink 删除文件出现的异常



怎样处理 unlink 删除文件出现的异常？
网络课堂：http://bbs.itzcn.com/thread-13625-1-1.html

当我使用 unlink 删除正在使用的文件时，总是会出现“拒绝访问”的提示。然后就想通过 try catch 语句捕捉这个错误。

请问，要怎么做才可以实现啊。

【解决办法】：出现“拒绝访问”提示，那是因为你所在的用户权限不够。如果有 FTP 权限的话，先把要删除文件的权限设置为 777，再执行 unlink 删除就可以了。

4.4.2　自定义异常处理提示的问题



自定义异常处理提示的问题。
网络课堂：http://bbs.itzcn.com/thread-13626-1-1.html

PHP 异常处理中，下面代码为什么运行不了？

```
<?php
try {
    throw new Exception('你处理的程序出现异常',42);
    catch (Exception $e){
    echo $e->getCode();
    }
}
?>
```

【解决办法】：这是 PHP 的一种错误订制方式，通过这个类来捕获相关的系统出错提示，再经过处理后输出给用户。要用 Exception 处理出错提示的话，得继承这个类后写一个具体如何处理出错信息的实现类，这样就可以了。

4.4.3　PHP 异常处理的 SQL 错误问题



PHP 异常处理的 SQL 错误问题？
网络课堂：http://bbs.itzcn.com/thread-13627-1-1.html

一个 try catch 的语句，当在 try 块中有错误时，cath 语句却并没有捕捉到，而是输出错误数组。这是为什么，如下是输出的结果。

```
MySQL server error report:Array ( [0] => Array ( [message] => MySQL Query Error )
[1] => Array ( [sql] => SELECT * FROM 'yhhg_goodspar' WHERE (1 >= 'g_s_price'
AND 1 <='s_price') ) [2] => Array ( [error] => Unknown column 'g_s_price' in
'where clause' ) [3] => Array ( [errno] => 1054 ) )
```

很明显是个数组，为什么不能捕获到这个错误。而我在 try 块中如果写个手工的 throw 却可以捕获得到，难道一定要手工抛出异常才行？

另外，请高手告诉我这个错误数组有什么办法可以获取到它。谢谢。

【解决办法】：SQL 错误是捕获不了的，因为 PHP 只要成功地把 SQL 命令提交给了数据库服务器就算成功，至于数据库里面执行是否成功，必须在每个语句之后检测，例如：

```
$db=mysql_connect(...);
if (!$db){
  echo '连接数据库失败,错误原因:'.mysql_error();
  exit;
}
```

再如：

```
$sql='...';
$res=mysql_query($sql);
if (!$res){
  echo '执行 SQL 语句错误，SQL 语句:'.$sql.'<br>错误原因:'.mysql_error();
  exit;
}
```

4.4.4　如何获取异常详细信息的问题



如何获取异常详细信息的问题?
网络课堂：http://bbs.itzcn.com/thread-13628-1-1.html

大家帮忙看看以下的代码：

```
if ($action == 'delete') {
   //删除译员测试信息
   delTranslatorTestInfo($db);
  } else {
   //显示译员测试信息
   displayTranslatorProfileInfo($db);
  }
} catch (Exception $e) {
  echo  $e-> getMessage();
}
```

执行后提示“parameter errors”，请问要获取更多异常信息，我应该要怎么做？

【解决办法】：

(1) 首先在所有 PHP 页面上添加“error_reporting(E_ALL ^ E_NOTICE);”语句。

(2) 然后把上述代码中的异常处理去掉。

(3) 另外，最好不要用 Exception 类。因为它是所有异常的父类，所以抛出的异常很笼统。要获取异常的详细信息，需要使用它的子类，或者是自定义异常类。

(4) 像你这类问题，应该是 SQL 语句出问题了。所以可以使用 mysql_error()来抛出错误，例如下面的例子。

```
$result=mysql_query($sql) or die(mysql_error());
```

4.5　习　　题

一、填空题

(1) __________指令是确定错误报告的敏感度级别。

(2) __________的作用会让 PHP 在$php_error_msg 变量中存储最近的错误信息。

(3) __________指令用于在程序运行时，防止 PHP 记录在同一文件中同一行上发生的重复错误信息。

(4) __________指令会显示 PHP 引擎初始化时遇到的所用错误。

(5) 基本异常类默认的构造函数是__________。

(6) __________指令是告诉 PHP 应当记录错误信息，并保存到某个文件或 syslog 中。

(7) __________指令用于 PHP 在忽略重复错误时将不考虑错误的来源，而且每一个错误只能记录一次。

(8) 基本异常类的重载构造函数可以有两个参数，一个是 message，另一个是__________。

(9) __________方法用于返回传递给构造函数的错误消息。

(10) __________方法用于返回出现错误时抛出异常的行号。

二、选择题

(1) 如何使用 PHP 输出"hello world"？__________

A．"Hello World"　　B．echo"Hello World"

C．Document.Write("Hello World")　　D．$a="Hello World"

(2) 如何从使用 get()方法提交的表单中获取数据？__________

A．$_GET[];　　B．Request.QueryString

C．Request.Form　　D．$_POST[]

(3) 在 PHP 中创建函数的正确方法是__________。

A．function myFunction()　　B．create myFunction()

C．new_function myFunction()　　D．new myFunction()

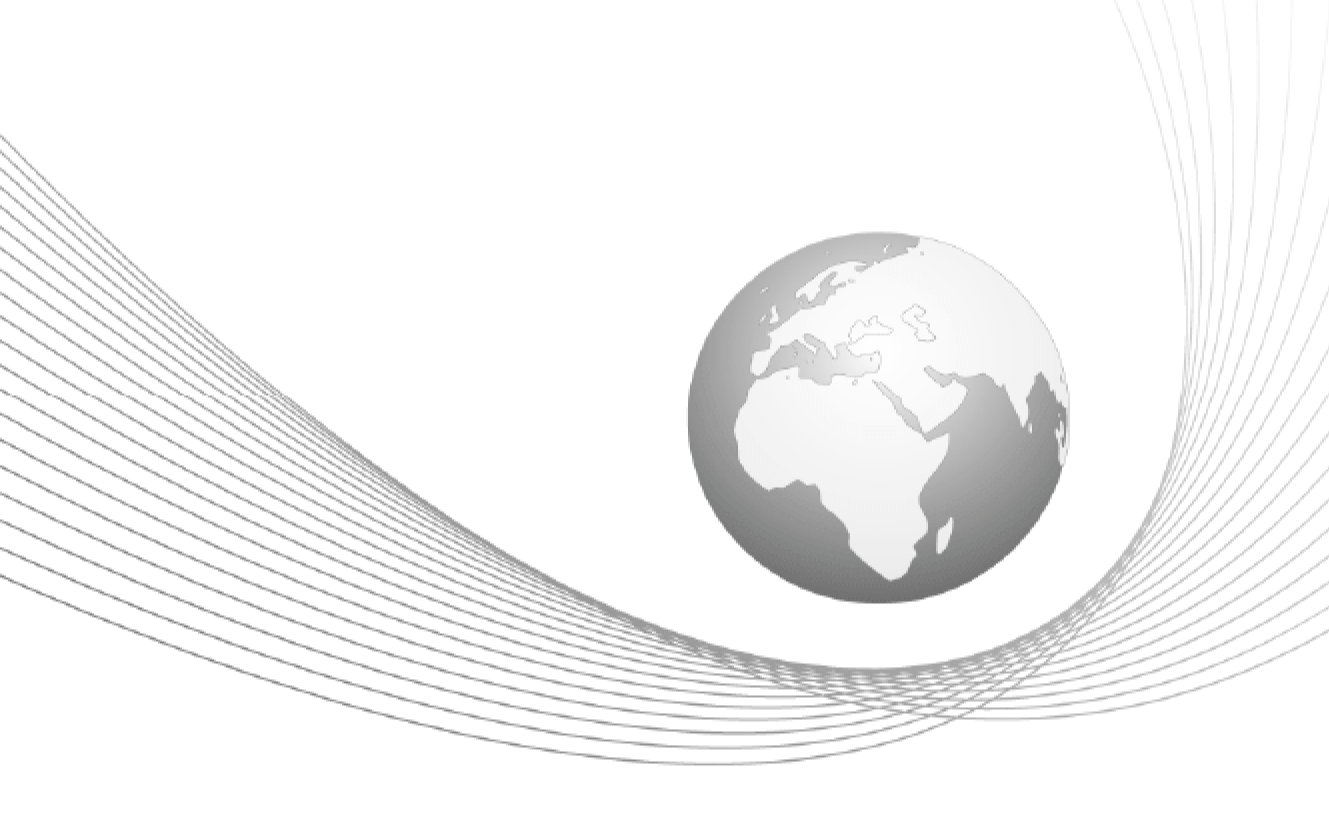
三、上机练习

上机练习： 自己动手制作注册页面，实现用户注册。

练习要求：首先制作用于用户注册信息的页面，然后定义一个自定义异常类 MyException 类，在该类中定义一个异常处理方法 doexception，对用户输入的信息进行异常处理。执行效果如图 4-3 所示。



图 4-3　注册页面



第 5 章　字符串与正则表达式

内容摘要：

字符串是程序开发中经常要使用到的重要内容之一，特别是在网络应用程序中，有相当多的内容都要与字符串发生联系。如：用户提交的内容为字符串、从库中查询出的结果为字符串。那么如何对字符串进行有效的操作，称为字符串处理的关键。本章将介绍 PHP 中常用字符串处理及与之相关的正则表达式应用。

学习目标：

- 复杂偏移语法。
- 确定字符串长度。
- 字符串比较函数。
- 字符串大小写转换。
- 字符串与 HTML 相互转换。
- 正则表达式中的特殊字符。
- 正则表达式中的预定义字符类。
- Perl 风格的正则表达式。
- 正则表达式函数。

5.1 PHP 中的偏移语法

PHP 是一种弱类型语言，它能够将字符串看作有意义的数组。因此，所有的字符串都可以看作一个连续的实体，同样也可以看作字符串集合，也就是说可以同时用两种方式输出同样的字符串，例如以下代码：

```
<?php
$thing="窗内网";
echo $thing;
echo "<br/>";
echo $thing[0];
echo $thing[1];
?>
```

上述代码看起来非常容易，但对初学者来说，很容易进入一个误区，在代码中要把这个数据当作字符串还要一个数组？如果采用这样松散的语法，将不能做到针对字符串运行时的代码优化，主要原因是这种脚本式引擎无法区分字符串和数组。因此，中括号偏移语法已经不用了，而是使用大括号语法，详细代码如下：

```
<?php
$thing="窗内网";
echo $thing;
echo "<br/>";
echo $thing{0};
echo $thing{1};
?>
```

虽然说中括号已经废弃不用了，但对于早期开发者来说它同样有价值。在开发过程中，我们建议使用大括号语法形成。

5.2 用 户 注 册

PHP 中有许多内置函数用于处理字符串，比如截取字符串、比较字符串长度是否相同等。在创建字符串之后，我们就可以对它进行操作了。可以直接在函数中使用字符串，或者把它存储在变量中。本节我们将对 PHP 中的字符串操作函数进行详细讲解。



视频教学： 光盘\videos\05\正则表达式匹配.avi　　长度：7 分钟
光盘\videos\05\preg_match.avi　　长度：4 分钟
光盘\videos\05\正则表达式替换.avi　　长度：5 分钟
光盘\videos\05\preg_replace.avi　　长度：6 分钟
光盘\videos\05\正则表达式分割.avi　　长度：7 分钟

5.2.1　基础知识——PHP 中的字符串处理

字符串是个特殊的字符数组。与普通字符数组不同的是，字符串在结尾处有一个字符\0，表示字符串的结束。在 PHP 中字符串有特殊的初始化方式和专门的处理函数。

1. 获取字符串的长度

有时需要判断用户输入的用户名信息长度是否合法，或者判断用户提交的某些信息是否在数据库字段指定的长度内，因此获取字符串的长度在程序中如影随形，在 PHP 中获取字符串的长度使用 strlen()函数，语法如下所示：

```
int strlen(string $string)
```

该函数返回值为一个整型，用来表示字符串的长度，如果返回值为 0，则表示该字符串为空。下面简单创建一个实例讲解如何使用 strlen()函数，详细代码如下所示：

```
//代码省略
<body style="font-size:12px">
<?php
    function GetLength($str){
    if(strlen($str)>10){
echo "你要测试的字符串长度大于 10，值为".strlen($str);
    }else{
    echo "你要测试的字符串长度小于 10，值为".strlen($str);
    }
    }
    GetLength("大家好！欢迎大家来到窗内网做客！");
?>
//代码省略
</body>
```

在上述代码中，使用 strlen()函数来获取 str 变量中的字符串长度，然后进行判断。如果长度大于 10，则输出“字符串长度大于 10…”的字样，如果字符串长度小于 10，则输出“字符串长度小于 10…”字样，空格同样也被计算在内。运行后页面会显示以下信息：

```
你要测试的字符串长度大于 10，值为 48
```

2. 字符串比较函数

对于那些学过 Java、C#和 ASP 编程语言的读者来说，比较两个字符串是否相同是编程中常用的功能之一，同样 PHP 中也为我们这些开发者设置了两个字符串之间的比较函数。其主要分为 4 种，下面就针对这 4 种 PHP 中字符串之间的比较函数进行详细讲解。

1)　strcmp()函数

strcmp()函数的主要功能是以区分大小写的方式比较两个字符串的大小，它的语法格式如下所示：

```
int strcmp(string str1,string str2);
```

strcmp()函数对字符串 str1 和 str2 进行二进制安全的比较，并且区分大小写，同时返回如

下 3 个值中的其中一个，如下。

- 返回 0 则表示 str1 和 str2 相等。
- 返回-1 则表示 str1 小于 str2。
- 返回 1 则表示 str1 大于 str2。

通过 strcmp()函数返回结果可以看出，strcmp()函数返回 int 类型数据。下面创建一个示例，具体讲解如何使用 strcmp()函数，以及在何时使用。

通常情况下该函数经常使用在用户注册功能中，它的主要作用是针对用户输入两次密码是否相同进行基本的验证。之所以输入两次密码，是因为有些用户会在不经意间错误输入密码或者大小写输入错误等原因导致密码不正确。所以使用 strcmp()函数比较两次密码是否相同非常合适。代码如下：

```
<?php
    if(isset($_POST['submit']))
        {
        if(strcmp($_POST['userPwd'],$_POST['twopwd'])==0)
        {
            echo "<hr/><li>恭喜你注册成功</li><br/>";
            echo "<li>你的用户名是：</li>".$_POST['userName'];
            echo "<br/><li>你注册的密码是：</li>".$_POST['userPwd'];
        }
        else
        {
            echo "<hr/> >你注册的密码和重复密码不同，请重新输入你的密码";
        }
    }
?>
```



对于 strcmp()，字符串必须完全匹配才认为是相等的。例如，Supersecret 不同于 supersecret。如果要以不区分大小写的方式来比较两个字符串，可以考虑下面介绍的 strcasecmp()函数。

针对这个函数，还有一个容易让人混淆的地方是两个字符串相等时要返回 0。这与使用==操作符完成字符串比较有所不同，例如以下代码。

```
if($str1==$str2)
```

在上述代码中，这种比较方式和使用 strcmp()函数的目的是相同的，但是一定要注意它们返回值是不同的，使用 strcmp()函数返回的是 0、-1 或 1，而使用“==”操作符对字符串进行比较返回的是 true 或 false。

2) strcasecmp()函数

strcasecmp()函数的作用与 strcmp()非常相似，只是 strcasecmp()在比较字符串时不区分大小写，该函数的语法格式如下所示：

```
int strcasecmp(string str1,string str2)
```

下面的例子要比较两个电子邮箱地址，对此使用 strcasecmp()很合适，因为电子邮箱的大

小写是不区分的。

```
<?php
    $str1 ="DUANSHAOZHI@126.com";
    $str2 = "duanshaozhi@126.com";
    if(strcasecmp($str1,$str2)==0)
        echo "不区分大小写";
    else
        echo "区分大小写";
?>
```

上述代码保存后运行结果将会输入“不区分大小写”字样，表示上述函数可以用来判断不区分大小写的比较。

3)　strspn()函数

strspn()函数用于找出指定字符串在另一字符串中存在的数目，该函数含有 4 个参数，其中第三个参数和第四个参数是可选的，分别表示开始匹配的位置和返回的最大值，其语法格式如下所示：

```
int strspn(string str1,string str2,int start, int length)
```

strspn()函数返回 str1 中包含有 str2 中字符串的第一部分的长度。下面的示例使用 strspn()判断一个密码不仅仅只由数字组成，代码如下所示：

```
<?php
$password="123456";
if(strspn(%password,"1234567890")==strlen($password))
echo "密码全部由数字组成";
?>
```

4)　strcspn()函数

strcspn()函数用于获取不匹配子字符串的长度。该函数的语法格式如下所示：

```
strcspn(string,char,start,length)
```

下面创建一个实例，具体说明如何运用 strcspn 函数，代码如下所示：

```
<?php
$password = "hn490714738";
if(strcspn($password,"1234567890")==0)
    echo "输入密码必须是字母数字组成！";
else
    echo"密码设置成功！";
?>
```

在上述代码中将不会出现提示信息，因为输入的用户密码是由字母和数字组合而成的字符串，如果输入密码仅为数字就要进行提示。

3. 字符串大小写转换

不同的语言将会用不同的函数进行字母的大小写转换，同样 PHP 中也有这样的函数来处

理字符串大小写转换，它们分别是 strtolower()、strtoupper()、ucfirst()和 ucwords()，表 5-1 列出了这四个函数的语法格式和功能。

表 5-1　大小写转换函数

名　称	语法格式	功　能
strtolower()	string strtolower(string $str)	把字符串的字母全部转换为小写字母
strtoupper()	string strtoupper(string $str)	把字符串的字母全部转换为大写字母
ucfirst()	string ucfirst(string $str)	如果字符串的第一个字符是字母，该函数则将其变成大写，非字母字符不受影响
ucwords	string ucwords(string $str)	该函数把字符串中每个单词的第一个字母变为大写，非字母字符不受影响

1)　strtolower()函数

strtolower()函数的语法格式如下所示：

```
string strtolower(string str)
```

strtolower()函数将 str 全部转换为小写字母，并返回修改后的字符串(非字母符号不受影响)。下面的示例使用 strtolower()将 URL 转换为小写字母，代码如下所示：

```
<?php
    $url="http://WWW.ITZCN.COM";
    echo strtolower($url);
?>
```

运行该段代码结果如下所示：

```
http://www.itzcn.com
```

2)　strtoupper()函数

strtoupper()函数的语法格式如下所示：

```
string strtoupper(string str)
```

用来将字符串全部转换为大写字母。下面代码是使用 strtoupper()函数将小写邮箱地址转换为大写字母：

```
<?php
    $mail="duanshaozhi@126.com";
    echo strtoupper($mail);
?>
```

上述代码运行后的结果如下所示：

```
DUANSHAOZHI@126.COM
```

3)　ucfirst()函数

ucfirst()函数的语法格式如下所示：

```
string ucfirst(string str);
```

如果字符串 str 的第一个字符是字母，可以使用 ucfirst()函数将其改为大写(非字母符号将不会受影响)。此外，字符串中的任何大写字母也保持不变。下面的代码是使用该函数将一句话首字母转换为大写：

```
<?php
    $str=" welcome to China!";
    echo ucfirst($str);
?>
```

上述代码运行后结果如下所示：

```
Welcome to China!
```



在使用 PHP 中的 ucfirst()函数时，第一个字母确实被大写，但是原来已经大写的单词将会保持不变。

4)　ucwords()函数

ucwords()函数的语法格式如下所示：

```
string ucwords(string str)
```

ucwords()函数将字符串中每个单词的第一个字母更改为大写。汉字或非字母符号将不受影响。下面使用 ucwords()函数将字符串中的每个单词首字母转换为大写，代码如下所示：

```
<?php
    $str="Welcome to the connection window! ";
    echo ucwords ($str);
?>
```

上述代码运行后结果如下所示：

```
Welcome to The Connection Window!
```

4. 填充和剔除字符串

在进行 Web 编程的过程中，因为程序的需要，需把字符串存在的一些字符从字符串中去掉，或者是在字符串中添加某些需要的字符串。为了实现该功能，PHP 提供了一些填充和剔除功能的函数。在这里将对这些函数进行介绍。

1)　ltrim()函数

ltrim()函数用来去除字符串左侧的空白或者指定的字符，在这里可以指定的字符包括空格、水平制表符(\t)、换行、回车(\r)、空值(\0)、垂直制表符(\x0b)等。要想去掉指定的某些字符串，就可用 ltrim()函数实现。该函数的语法格式如下所示：

```
string ltrim ( string $str [, string $charlist] )
```

下面通过使用 ltrim()函数实现去除字符串左侧的空白或者指定的字符，代码如下所示：

```
<?php
$str = " \t\t 欢迎大家来到窗内网！ ";
$out1 = ltrim($str);
echo $ out1;
```

```
echo "<hr>";
$out2= ltrim($str, " \t.");
echo $out2;
echo "<hr>";
$out3 = ltrim($str, "窗内网");
echo $out3;
?>
```

运行上述代码，结果如下所示：

```
欢迎大家来到窗内网!
-----------------------------------------------
欢迎大家来到窗内网!
-----------------------------------------------
欢迎大家来到
```

2) rtrim()函数

rtrim()函数和ltrim()函数功能相同，只是它从字符串的右侧删除字符，同样该函数也可以替换某些指定字符。rtrim()函数的语法格式如下所示：

```
string rtrim(string str[,string charlist])
```

该函数的使用示例如下所示：

```
<?php
$str = "欢迎大家来到窗内网！ ";
echo $str."<br/>";
echo rtrim($str);
?>
```

执行该段代码后，输出的结果中第一行字符串最后有空格，而第二行字符串则没有空格，结果如下所示：

```
欢迎大家来到窗内网!
欢迎大家来到窗内网!
```

3) trim()函数

trim()函数的语法格式如下所示：

```
string trim(string str[,string charlist])
```

trim()函数是ltrim()函数和rtrim()函数的组合，凡是ltrim()函数和rtrim()函数可以实现的功能该函数都可以做到。该函数可以从字符串的两侧删除指定字符。该函数的使用示例如下所示：

```
<?php
echo trim(" 欢迎大家来到窗内网！ ","大家");
?>
```

上述代码的运行结果中字符串的两端空格都会被删除，而且指定的字符串也被删除，结果如下所示：

```
欢迎来到窗内网
```

4)　str_pad()函数

str_pad()函数的语法格式如下所示：

```
string str_pad(string str,int length[,string pad_string[,int pad_type]])
```

str_pad()函数的作用是把字符串填充为指定的长度，该函数有 4 个参数：第一个参数规定要填充的字符串；第二个参数规定新字符串的长度，如果该值小于原始字符串的长度，则不进行任何操作；第三个参数可选，规定供填充使用的字符串，默认是空白；第四个参数可选，规定填充字符串的哪边，有 STR_PAD_RIGHT、STR_PAD_LEFT 和 STR_PAD_BOTH 三个常量可以选择，分别表示右补白，左补白和两侧补白，默认值为 STR_PAD_RIGHT。下面的示例展示了如何使用 str_pad()函数填充字符串，代码如下：

```
<?php
echo str_pad("开始",10). "结束"
?>
```

上述代码返回结果如下所示：

```
"开始" "结束"
```

下面实例则是使用 str_pad()的可选参数，代码如下所示：

```
<?php
    $header="窗内网";
    echo str_pad($header,15,"=+",STR_PAD_BOTH);
?>
```

上述代码的运行结果如下所示：

```
=+=+窗内网=+=+=
```



如果在之前达到了 length 的字符长度，str_pad()将截断由 pad_string 定义的字符串模式。

5.2.2　基础知识——字符和单词计数

开发过程中，要确定某些字符串的字符或者单词的个数是比较常用的操作。PHP 在字符串解析方面提供了强大的功能，使得这些任务非常简单。接下来就介绍一下字符和单词计数这两个函数。

1. count_chars()函数

count_chars()函数主要用于返回字符串所用字符的信息，可使用多种模式返回结果。可选参数 mode 默认值为 0，并且函数的行为也依赖于该参数。该函数语法格式如下所示：

```
mixed count_chars (string $string [, int $mode])
```

在使用的过程中，count_chars()函数的参数 mode 值不同时，函数所表达的功能也不相同。以下是参数 mode 的可取值：

- 0：以字符作为键名、出现次数作为键值的数组。

- 1：与 0 相同，但只列出出现次数大于零的字节值。
- 2：与 0 相同，但只列出出现次数等于零的字节值。
- 3：返回由所有使用了的字节值组成的字符串。
- 4：返回由所有未使用的字节值组成的字符串。

下面创建一个示例，具体说明如何运用 count_chars()函数，代码如下所示：

```
<?php
$data = "欢迎大家来到窗内网！热烈欢迎！";
$chart = count_chars($data,1);
foreach ($chart as $s=>$num)
{
    echo "字符“".chr($s)."”出现".$num."次<br/>";
}
?>
```

执行该段代码，执行结果如下所示：

```
字符“欢”出现 2 次
字符“迎”出现 2 次
字符“大”出现 1 次
字符“家”出现 1 次
字符“来”出现 1 次
字符“到”出现 1 次
字符“窗”出现 1 次
字符“内”出现 1 次
字符“网”出现 1 次
字符“!”出现 2 次
字符“热”出现 1 次
字符“烈”出现 1 次
```

2．str_word_count()函数

str_word_count()函数的作用是获取字符串里面的英文单词总数的信息，该函数语法格式如下所示：

```
mixed str_word_count (string $string [, int $format [, string $charlist]])
```

其中，第一个参数表示要检查的字符串。第二个参数可选，规定 str_word_count()函数的返回值。如果定义了第二个参数，那么参数值不同，函数所表示的意义也不同，省略的默认值是 0。具体介绍如下。

- 0：表示只返回单词的个数。
- 1：表示将找到的单词作为一个数组返回。
- 2：表示找到的单词和单词所在的字符位置信息作为一个联合数组返回。

第三个参数也是可选参数，表示需要忽略的断词符号，默认的断词符号有空格和数字等。下面创建一个示例，具体讲解如何使用该函数，代码如下所示：

```
<?php
```

```
$summary ="Welcome to the connection window";
$word = str_word_count($summary);
echo $summary."句中共有".$word."个单词";
echo "<br/>详细如下所示：";
$words = str_word_count($summary,2);
$frequency = array_count_values($words);
print_r($frequency);
?>
```

上述代码中使用了 arry_count_values()函数，执行结果如下所示：

```
Welcome to the connection window 句中共有 5 个单词
详细如下所示：Array ([Welcome] => 1 [to] => 1 [the] => 1 [connection] => 1 [window]
=> 1)
```

5.2.3　基础知识——字符串与 HTML 相互转换

PHP 有一个非常强大的功能，可以使用一些函数把字符串或整个文件转换为适合于 Web 形式的文本信息。而且 PHP 还提供了两种这样的函数：第一种是将纯文本转换为 HTML，第二种是将 HTML 转换为纯文本，以下详细介绍。

1. 将纯文本转换为 HTML

在 PHP 中要想将纯文本转换为 HTML 的信息，就需要使用 nl2br()、htmlentities()、get_html_translation_table()和 strtr()函数，下面以示例的方式详细讲解这些函数。

1)　nl2br()函数

nl2br()函数在字符串中的每个新行(\n)之前插入 HTML 换行符(
)，并返回修改后的字符串。该函数的语法格式如下所示：

```
string nl2br ( string $string )
```

下面通过一个简单的实例讲解如何使用 nl2br()函数，代码如下所示：

```
<?php
 $str= "大家好！\n 欢迎大家来到窗内网，免费学习编程！";
 $s = nl2br($str);
 echo $s;
?>
```

执行该实例，查看页面源代码，结果如下所示：

```
大家好！<br />
欢迎大家来到窗内网，免费学习编程！
```

2)　htmlentities()函数

htmlentities()函数可以把字符串中的一些字转换为 HTML 实体，默认情况下主要包括 4 个字符：“<”、“>”、“&”和“"”，该函数语法格式如下所示：

```
string htmlentities ( string $string [, int $quote_style [, string $charset]] )
```

htmlentities()函数的第二个可选参数规定如何编码单引号和双引号，可以选择三个常量：ENT_COMPAT 表示对双引号进行编码，不对单引号进行编码；ENT_QUOTES 表示同时对单引号和双引号进行编码；ENT_NOQUOTES 表示两个都不转换，默认值为 ENT_COMPAT。第三个可选参数表示所转换字符的编码集，其常用编码如表 5-2 所示。

表 5-2　字符编码集

字 符 集	别　名	描　述
ISO-8859-1	ISO8859-1	西欧，Latin-1
ISO-8859-15	ISO8859-15	西欧，Latin-9。增加了 Latin-1(ISO-8859-1)中缺少的欧元符号、法国及芬兰字母
UTF-8	—	ASCII 兼容多字节 8-bit Unicode
cp1252	Windows-1252, 1252	Windows 对于西欧特有的字符集
KOI8-R	koi8-ru, koi8r	俄文。PHP 4.3.2 开始支持该字符集
BIG5	950	繁体中文，主要用于中国台湾
GB2312	936	简体中文，国际标准字符集
BIG5-HKSCS	—	繁体中文，Big5 的延伸，主要用于香港
Shift_JIS	SJIS, 932	日文
EUC-JP	EUCJP	日文

下面创建一个示例，具体讲解如何使用该函数转换语句，代码如下所示：

```
<?php
$str = "<title>Welcome to 中国</title>";
echo htmlentities($str);
echo "<br/>";
echo htmlentities($str,ENT_NOQUOTES,GB2312);
?>
```

通过运行上述代码，可以看到在页面显示的形式与定义时的字符串相同，此时查看源代码，结果如下所示：

```
&lt;title&gt;Welcome to &Ouml;&ETH;&sup1;&uacute;&lt;/title&gt;<br/>
&lt;title&gt;Welcome to 中国&lt;/title&gt;
```

3)　htmlspecialchars()函数

htmlspecialchars()函数将一些预定义的字符转换为 HTML 实体，其语法格式如下所示：

```
htmlspecialchars(string,quotestyle,character-set)
```

此函数只转换下面的特殊字符，分别是：“&(和)”转成“&amp”、“"(双引号)”转成“"”、“<(小于)” 转成 “<” 、 “>(大于)” 转成 “&gt” 。该函数第二个参数和第三个参数的含义与 htmlentities()函数中的参数含义相同。下面的代码为该函数的使用示例：

```
<?php
$str = "'今天温度'>'昨天温度'";
echo htmlspecialchars($str, ENT_COMPAT);
```

```
echo "<br />";
echo htmlspecialchars($str, ENT_QUOTES);
echo "<br />";
echo htmlspecialchars($str, ENT_NOQUOTES);
?>
```

运行上述代码，查看其结果确认页面的源码如下所示：

```
'今天温度'&gt;'昨天温度'<br />
&#039;今天温度&#039;&gt;&#039;昨天温度&#039;<br />
'今天温度'&gt;'昨天温度'
```



如果 gethtmlspeicalchars() 函数要与 nl2br() 等函数结合使用，应当在 gethtmlspeicalchars()之后再执行 nl2br()。否则，nl2br()生成的
标记将被转换为可见字符。

4)　get_html_translation_table()函数

使用 get_html_translation_table()函数是将文本转换为等价形式的一个遍历函数，该函数返回由被 htmlentities()和 htmlspecialchars()函数使用的翻译表，该函数的语法格式如下所示：

```
get_html_translation_table(function,quotestyle)
```

get_html_translation_table()函数有两个常量参数。第一个参数规定返回哪个翻译表，默认是 HTML_SPECIALCHARS，该参数包括两种模式：HTML_ENTITIES 表示大范围的 htmlentities()函数所用到的转换内容；HTML_SPECIALCHARS 表示小范围的 htmlspecialchars()函数所用到的转换内容。第二个参数用来定义如何对单引号和双引号进行编码，该参数有以下 3 种模式：ENT_COMPA、ENT_QUOTES 和 ENT_NOQUOTES。下面创建一个使用 get_html_translation_table()函数的示例，代码如下所示：

```
<?php
print_r (get_html_translation_table());
echo "<br />";
print_r (get_html_translation_table(HTML_ENTITIES));
?>
```

5)　strtr()函数

strtr()函数用来转换字符串中的特定字符，其语法格式如下所示：

```
string strtr ( string $str, string $from, string $to )或
string strtr ( string $str, array $replace_pairs )
```

第一种代码格式中该函数共有 3 个参数，替换时会将第二个参数的字符串替换为第三个参数的字符串，如果两个字符串参数的长度不一致，则较长的那个字符串将会被截断。第二种代码格式中使用两个参数，其中第二个参数是一个准备进行替换处理的数组。下面用该函数创建一个示例，以第一种代码格式为例的代码如下所示：

```
<?php
echo strtr("my name is duanshaozhi! ","duanshaozhi","DuanShaozhi");
?>
```

执行代码，结果如下所示：

```
my name is DuanShaozhi!
```

2. 将 HTML 转换为纯文本

有些情况下，需要将 HTML 文件转换为纯文本，因此 PHP 中的一些函数可以把 HMTL 标记的信息转换为纯文本信息，下面这个函数就可以完成这项功能。

strip_tags()函数的作用为剥去 HTML、XML 以及 PHP 的标签，语法如下所示：

```
strip_tags(string,allow)
```

strip_tags()函数经常用在论坛或者文章显示页面中，用来过滤一些标签的出现。其中第二个参数表示规定允许的标签，这些标签不会被删除。如果字符串的 HTML 及 PHP 标签原来就有错，则返回错误。该函数的使用示例如下所示：

```
<?php
echo strip_tags("新闻<b><i>新闻标题</i></b>","<b>");
?>
```

运行以上代码，结果如下所示：

```
新闻 新闻标题
```

通过运行查看源码可以发现，运行后结果没有 HTML 标签。

5.2.4 实例描述

该实例主要通过使用 PHP 的一些函数来实现用户注册中判断用户提交信息是否合法等功能的。使用 PHP 中的 strlen()函数来判断用户提交的信息是否为空，然后通过使用字符串大小写转换函数将用户两次输入的密码全部转换为小写字母，用来实现不区分大小写的功能。

5.2.5 实例应用

【例 5-1】 用户注册

首先创建用户注册界面，添加用户名、用户密码、确认密码、邮箱等用户信息，代码如下所示：

```
    <form action="registerservice.php" method="post" id="apForm">
        <input type="hidden" name="referer" value="./index.html" />
        <div class="content">
        <p>
        <label for="email"> 您的登录名：</label>
    <input class="input" type="text" id=username maxlength=32  name="username" />
        </p>
        性别：
        <label>
    <input type="radio" name="RadioGroup1" value="男性" id="RadioGroup1_0" />
```

```
        男性</label>
        <label>
        <input type="radio" name="RadioGroup1" value="女性" id="RadioGroup1_1" />
        女性</label>
<p>
        <label for="email">登录密码：</label>
        <input class="input" style="width:148px;" type=password maxlength=16
id="password" name="userpwd1" />
        </p>
        <p>
            <label for="email"> 密码确认：</label>
            <input class="input" style="width:148px;" type=password maxlength=16
id="password2" name="userpwd2" /></p>
        <p>
            <label for="email"> 电子邮箱：</label>
            <input id=email maxlength=32  class="input" name=email />
        </p>
            <INPUT type="hidden" name="registersubmit" value="true">
            <INPUT type="submit" name="mysubmit" value="" class="tj">
            </p>
        </div>
     </form>
```

上述代码中，设置了用户名、密码和邮箱等文本域，当用户单击【注册】按钮时则会使用 form 表单提交至 registerservice.php 页面进行处理，该页面的代码如下所示：

```
<?php
    $name=$_POST["username"];
    $userpwd1=$_POST["userpwd1"];
    $userpwd2=$_POST["userpwd2"];
    if(strlen($name)>15||strlen($name)<1)
    {
        echo "输入用户名长度不符合！";
    }
    if(strlen($userpwd1)>1)
    {
    if(strcmp(strtolower($_POST['userpwd1']),strtolower($_POST['userpwd2']))!=0)
    {
        echo '两次密码不符合！';
    }
    else
    {
    echo
    '申请成功！';
    }
    }else
    {
        echo '请输入登录密码，不区分大小写！';
    }
?>
```

上述代码中，使用$_POST 获取用户表单中提交的数据，表单中的数据以数组形式保存，我们使用数组获取数据形式获取每个元素的数据，判断提交数据是否合法，如果两次数据不相同，则提示用户两次密码不符合，当内容为空则提示用户重输入用户密码等信息，运行结果如图 5-1 所示。

5.2.6　运行结果

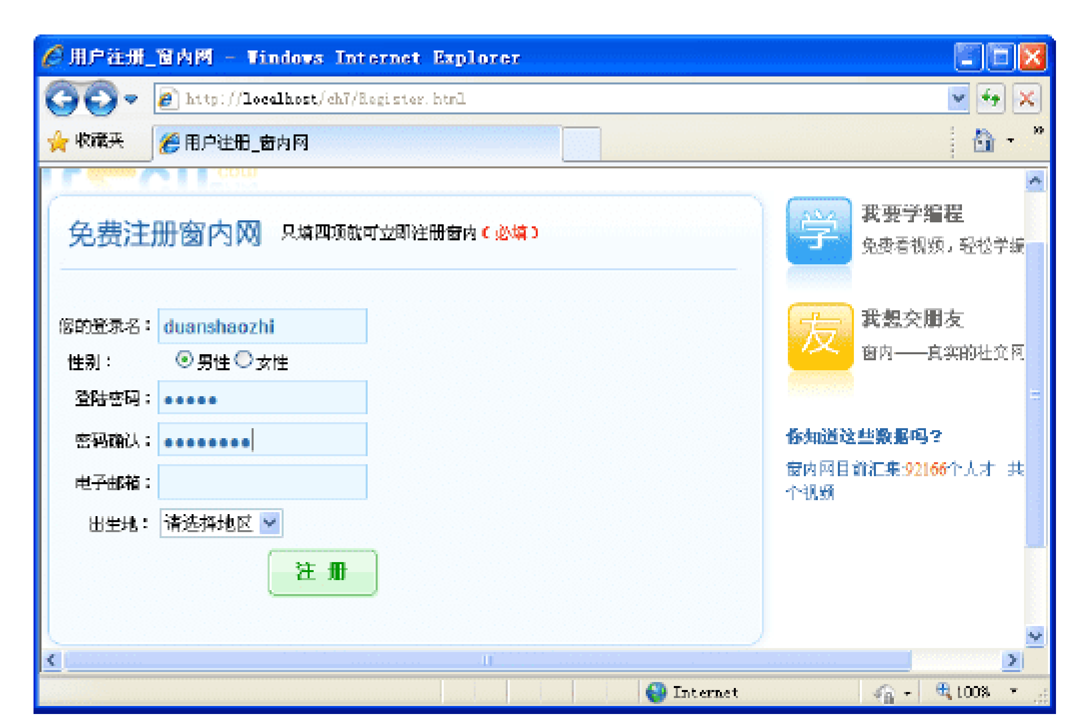
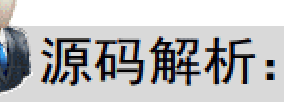




图 5-1　用户注册功能

5.2.7　实例分析



源码解析：

上述实例中使用 strlen()函数判断字符串的长度，从而判断字符串是否为空，当内容不为空时，使用 strtolower ()函数将输入的大写字母转换为小写字母，最后使用 strcmp()函数判断两次输入密码是否相同。

5.3　用户登录验证

在正则表达式基础上，可以根据预定义的语法表述或匹配数据。正则表达式本身只是字符串模式，用于匹配某些文本。这个字符串可以是您所熟悉的模式，如单词 dog，也可以是模式匹配上下文中特殊意义的模式。

PHP 提供了特定于两组正则表达式功能的函数，分别对应于一类正则表达式：POSIX 风格和 Perl 风格。每种正则表达式都有自己独特的语法风格，我们将在后面各节中分别讲解。

视频教学： 光盘\videos\05\中括号.avi　　长度：10 分钟
光盘\videos\05\特殊字符.avi　　长度：12 分钟
光盘\videos\05\预定义字符类.avi　　长度：5 分钟
光盘\videos\05\Perl 正则表达式.avi　　长度：5 分钟
光盘\videos\05\修饰符.avi　　长度：5 分钟

5.3.1　基础知识——正则表达式中的特殊字符

在使用正则表达式时，上下文中使用中括号时，有一种特殊的作用—— 查询定界范围内的字符。

中括号内字符序列和单个字符的出现频率或者位置可以使用一个特殊的字符来代替，这些特殊的字符都有不同的含义，其中+、*、?、{occurrence_range}和$等标志都出现在一个字符串序列的后面，表 5-3 是一些常用的特殊字符。

表 5-3　特殊字符

常用形式	说　明
p+	匹配任何一个至少包含 p 的字符串
p*	匹配任何包含零个或多个 p 的字符串
p?	匹配任何包含零个或一个 p 的字符串
p{2}	匹配任何包含两个 p 序列的字符串
p{2,3}	匹配任何包含两个或三个 p 序列的字符串
p{2.}	匹配任何至少包含两个 p 序列的字符串
p$	匹配任何以 p 结尾的字符串
^p	匹配任何以 p 开头的字符串
[^a-zA-Z]	匹配任何不包含从 a 到 z 和从 A 到 Z 的字符串
p.p	匹配任何包含字符 p、接下来是任何字符串、再接下来又是 p 的字符串
^.{2}$	匹配任何只包含两个字符的字符串
(.*)	匹配任何被和包围的字符串
p(hp)*	匹配任何包含一个 p，后面是零个或多个 hp 的字符串
(a|b)	匹配任何包含 a 或者 b 的字符串

通过使用特殊字符的正则表达式来处理一些字符串是非常方便的操作，可以通过使用正则表达式来指定输入内容的字符串格式。



在使用正则表达式时，默认情况下是区分大小写的，如果想让其不区分大小写，就可以使用 eregi()函数。

5.3.2　基础知识——正则表达式中的预定义字符类

通常在使用 PHP 正则表达式时，为了方便程序员的使用和开发，还可以使用一些预定义的字符范围，称为字符类。字符类可以指定整个字符范围，例如字母或整数集。预定义字符类如表 5-4 所示。

表 5-4　预定义字符类

类	描　述	扩　展
[:alnum:]	字母和数字字符	[0-9a-zA-Z]
[:alpha:]	字母字符(字母)	[a-zA-Z]
[:ascii:]	7 位 ASCII	[\x01-\x7F]
[:blank:]	水平空白符(空格、制表符)	[\t]
[:cntrl:]	控制字符	[\x01-\x1F]
[:digit:]	数字	[0-9]
[:graph:]	用墨水打印的字符(非空格、非控制字符)	[^\x01-\x20]
[:lower:]	小写字母	[a-z]
[:print:]	可打印字符(图形类加空格和制表符)	[\t\x20-\xFF]
[:punct:]	任意标点符号，如句点(.)和分号(;)	[-!"#$%&'( )*+,./:;<=>?@[\\\]^_'{|}~]
[:space:]	空白(换行、回车、制表符、空格、垂直制表符)	[\n\r\t \x0B]
[:upper:]	大写字母	[A-Z]
[:xdigit:]	十六进制数字	[0-9a-fA-F]

由表 5-4 可以得知，每一个类都可以被用于替代一个字符类中的字符。例如：

```
[@[:digit:][:lower:]]
```

上面的代码是一个用来查询任意数字、小写字母的正则表达式。一个类不能反过来当作一个字符类使用，例如下面的代码就是一个错误的正则表达式。

```
ereg('[A-[:lower:]]', 'string');
```

一些地区把某些字符序列当作一个单独的字符来考虑，此时它们被称为排序序列(collating sequence)。在字符类中匹配这些多字符序列中的一个时，要把它用“[.”和“.]”括起来。例如，如果某地区有排序序列 ch，可以使用下面的字符类来匹配 s、t 或 ch：

```
[st[.ch.]]
```

例如，在文件中编写如下代码，它们会返回不同的结果内容：

```
<?php
echo ereg('^[:upper:]', 'itzcn');                    // 返回 false
?>
```

运行以上代码则返回 false，因为所指定的类表示大写字母，而指定的内容值为小字母。

5.3.3　基础知识——Perl 风格的正则表达式

Perl 是一种非常强大的解析语言之一，它为程序人员设计了一套非常全面的正则表达式语言，即使是最复杂的字符串模式，也可以用这种正则表达式语言搜索和替代。PHP 开发人员认

识到，与其重新开发正则表达式，不如让 PHP 用户之间使用非常流行的 Perl 正则表达式语言。

所谓 Perl 风格的正则表达式实际上相当于 POSIX 实现派生的，它们之间有着非常相似的语言风格，因此二者有很多类似的地方。下面先来看一个基于 Perl 正则表达式的简单例子：

```
/food/
```

字符串 food 放在两个斜线之间。与 POSIX 正则表达式类似，可以通过两次构建更复杂的字符串表：

```
/fo+/
```

上述代码将匹配“fo”字符串之后面跟一个或多个字符串。对于这个模式，可能的匹配包括 food、fool 和 fo4 等，下面是使用两次的另一个例子。

```
/fo{2,4}/
```

在将匹配 f 后面跟有 2～4 个“o”。对此，可能的匹配有 fool、fooool 和 foosball 等。

1. 修饰符

在编程的过程中，可能希望调整某个正则表达式的解释，也可能希望正则表达式完成不区分大小写的搜索或忽略语法中的注释。这些调整称为修饰符，修饰符对编写简单短小的表达式有很大的帮助，表 5-5 中列出了几个常用的修饰符。

表 5-5　修饰符

修 饰 符	描　述
i	完成不区分大小写的搜索
g	完成全部搜索
m	将一个字符串视为多行。默认情况下，^和$字符匹配字符串中的最开始和结尾。使用 m 修饰符将使^和$匹配字符串中每行的开始
s	将一个字符串视为一行，忽略其中所有换行字符。与 m 修饰符使用相反
x	忽略正则表达式中的空白和注释
u	第一次匹配后停止

修饰符一般直接放在正则表达式的后面，例如：

```
/ab/i
```

正则表达式/ab/原本只可以匹配字符串 ab，而在使用了修饰符 i 后，就可以匹配字符串 Ab、aB、AB 或 ab。

多个修饰符可以同时使用，例如：

```
/ab/is
```

修饰符可以直接放在表达式的后面，例如以下示例。

- /wnd/i：可以匹配 WMD、Wmd、WMd、wmd，以及字符串 wnd 的任何其他大写形式。

- /taxation/gi：由于不区分大小写，所以会查找到单词 taxation 的所有出现。可以使用这个全局修饰符计算出现的次数，或者结合替换特性用某些字符串代替所有出现的模式。

2．元字符

通过使用 Perl 正则表达式还可以实现其他功能，如使用各种元字符来搜索匹配。元字符就是前面有反斜线的字母字符，表示某种特殊含义。表 5-6 中是常用的元字符。

表 5-6　元字符

字　符	描　述
\A	只匹配字符串开头
\b	匹配单词边界
\B	匹配除单词边界之外的任意字符
\d	匹配数字字符，它与[1-9]相同
\D	匹配非数字字符
\s	匹配空白字符
\S	匹配非空白字符
[]	包围一个字符类
()	包围一个字符分组或定义一个反引用
$	匹配行尾
^	匹配行首
.	匹配除换行之外的任何字符
\	引出下一个元字符
\w	匹配任何只包含下划线和字母数字字符的字符串
\W	匹配没有下划线和字母数字字符的字符串

每个元字符只能匹配一个字符，如果要匹配多个，可以使用 POSIX 正则表达式中的特殊字符。

POSIX 正则表达式还可以用来匹配多个数字字符。表 5-6 中的常用元字符都比较好理解，下面通过一行代码来讲解如何使用 Perl 正则表达式：

```
^[a-zA-Z]([a-zA-Z0-9]*[-_.]?[a-zA-Z0-9]+)+@([\w-]+\.)+[a-zA-Z]{2,}$
```

上述代码是一个用来验证邮箱的正则表达式，邮箱名必须是字母和数字组合而成，接下来是一个@符号，紧接着是匹配任何甚至包含下划线和任意字母、数字字符的字符串，后面是字母字符。

3．定界符

定界符的主要作用是用来限定正则匹配表达式的左右界限符号，左右要求完全一致，表达式应被包含在定界符中，常用的符号有“/”和“@”等，任何不是字母、数字或反斜线“\”的字符都可以作为定界符。如果作为定界符的字符，那么就要使用反斜线来转换。

例如下面的字符串：

```
D://ch1/phpFile
```

可以使用以下的正则表达式匹配该字符串：

```
/D:\/\/ch1\/phpFile/
```

由于定界符使用的是“/”，而字符串中也包含有“/”，所以在正则表达式中，除了定界符以外，其他“/”前面都需要添加反斜线进行转义。这时，为了方便，我们最好选择使用其他字符作为定界符，例如以下正则表达式就是使用字符“#”作为定界符，这样就不用在“/”前面添加反斜线进行转义。

```
#D:/xiaoqi/phpFile#
```

5.3.4　基础知识——PHP 的正则表达式函数

PHP 为使用 Perl 兼容的正则表达式搜索字符串提供了 7 个函数：preg_grep()、preg_match()、preg_match_all()、preg_quote()、preg_replace()、preg_replace_callback()和 preg_split()函数。这些函数为我们提供了不同的搜索功能，下面将对这些函数进行详细的讲解。

1. preg_grep()函数

preg_grep()函数的语法格式如下所示：

```
array preg_grep(string pattern,array input[,flags])
```

preg_prep()函数用来搜索数组 input 中的所有元素，返回由与 pattern 匹配的所有元素组成的数组。下面示例使用这个函数在数组中搜索以 a 开头的单词：

```
<?php
    $words=array("dendrite","cant","alb","apple");
    $word=preg_grep("/^a/",$foods);
    print_r($food)
?>
```

运行上述示例，将会返回如下内容：

```
Array([0]=> alb [3]=> apple)
```



输出数组对应于输入数组的索引顺序。如果某个索引位置的值与模式匹配，这个值将被包含到输出数组的对应位置处；否则，该位置为空。如果希望删除数组中的空实例，可以通过使用函数 array_values()函数输出数组。

2. preg_match()函数

preg_match()函数语法结构如下所示：

```
int preg_match(string pattern,string string[,array matches][,int flags[,int
offset]]);
```

preg_match()函数在 string 中搜索 pattern 进行正则匹配，如果存在则返回 true，否则返回

false。可选的输入参数 matches 可以包含搜索模式中的子模式。下面通过一个简单的实例使用 preg_match()完成一个不区分大小写的搜索功能，代码如下所示：

```
<?php
$pattern = "/\ba\b/";
$string1 = "I am a student";
if(preg_match($pattem,$string1))print "通过认证！"
?>
```

执行结果如下所示：

```
通过认证！
```

3. preg_match_all()函数

preg_match_all()函数用于匹配字符串在模式中的所有出现结果，还可以通过可选的输入参数指定匹配顺序，并将每次出现结果放在数组中。字符串中匹配模式的所有出现，方便通过可选的输入参数指定的顺序，将每次出现放在数组中。preg_match_all()函数的语法格式如下：

```
int preg_match_all(string pattern, string string, array regs [, int order])
```

preg_match_all()函数的 regs 参数为必选参数。另外，可选参数 order 可以接收如下两个可选值。

- PREG_PATTERN_ORDER：选用此值时，$regs[0]是模式完全匹配的所有字符串的数组，$regs[1]是包含与第一个小括号内的正则表达式相匹配的所有字符串的数组，$regs[2]是包含与第二个小括号内的正则表达式相匹配的所有字符串的数组，以此类推。
- PREG_SET_ORDER：选用此值与选用 PREG_PATTERN_ORDER 稍有不同，$regs[0]是包含与第一个小括号内的正则表达式相匹配的所有字符串的数组，$regs[2]是包含与第二个小括号内的正则表达式相匹配的所有字符串的数组，以此类推。
  下面以验证电话号码格式是否正确为例，代码如下所示：

```
<?php
$telephone="0371-485585";
if (preg_match_all ("/(\d{4})-(\d{1,7})/", $telephone, $regs))
   {
   echo $regs[0][0].">>区号".$regs[1][0]."电话号".$regs[2][0];
}
else
{
    echo "不正确电话号码:".$telephone;
}
?>
```

- 执行结果如下所示：

```
0371-485585>>区号 0371 电话号 485585
```

4. preg_quote()函数

preg_quote()函数用于返回某个字符串的 Perl 正则表达式形式，其语法格式如下：

```
string preg_quote(string str [, string delimiter])
```

preg_quote()以 str 为参数并给其中每个属于正则表达式语法的字符前面加上一个反斜线。如果需要以动态生成的字符串作为模式去匹配，则可以用此函数转义其中可能包含的特殊字符。

如果提供了可选参数 delimiter，该字符也将被转义。delimiter 参数还可以用来转义 PCRE 函数所需要的定界符，最常用的定界符是斜线“/”，这些特殊字符主要包括：!、$、^、*、(、)、=、+、{、}、[、]、|、\、:、<、>。

下面的例子使用 preg_quote()函数实现正则匹配，代码如下所示：

```
<?php
$string = "[Welcome to the connection window!]";
$str1= preg_quote($string);
$str2 = preg_quote($string, 'w');
echo $str1;
echo '</br>';
echo $str2;
?>
```

执行结果如下所示：

```
\[Welcome to the connection window\!\]
\[Welcome to the connection \windo\w\!\]
```

通过以上例子可知，在使用 preg_quote()函数时如果没有指定定界符，在生成的正则表达式中，只会在特殊字符“!”、“[”和“]”前面添加反斜线。如果指定了定界符，生成的正则表达式中将会在定界符前面也添加反斜线，同时定界符还区分大小写。

5．preg_replace()函数

preg_replace()函数以指定的规则来解析比对指定的字符串，并将匹配的项替换为规定长度的字符串，该函数的语法格式如下：

```
string preg_replace(mixed pattern, mixed replacement, mixed subject [, int limit])
```

在 subject 中搜索 pattern 模式的匹配项并替换为 replacement。如果指定了 limit，则仅替换 limit 个匹配，如果省略 limit 或者其值为-1，则所有的匹配项都会被替换。

preg_replace()的每个参数(除 limit)都可以是一个数组。如果 pattern 和 replacement 都是数组，将以其键名在数组中出现的顺序来进行处理，但这不一定和索引的数字顺序相同。如果想使用索引标识哪个 pattern 将被哪个 replacement 来替换，应该在调用 preg_replace()之前用 ksort()对数组进行排序。

下面将通过 preg_replace()函数来替换一句话中的一个字母。

```
<?php
$string = "I am a teacher";
$pattern = array("/\bi\b/i", "/\bam\b/");
$replacement = array("He", "was");
$result = preg_replace($pattern, $replacement, $string);
echo $result;
?>
```

执行结果如下所示：

```
He was a teacher
```

6. preg_replace_callback()函数

preg_replace_callback()函数执行正则表达式搜索和替换，并返回用于替换的字符串。该函数有 5 个参数。第一个参数 pattern 表示该模式的搜索，它可以是一个字符串或一个字符串数组。第二个参数 callback 表示一个字符串回调将被调用，通过主题匹配的数组中的元素，回调应该返回替换字符串。第三个参数 subject 表示用字符串或一个字符串数组来搜索和替换。第四个参数 limit 表示对于每个模式每个科目字符串的最大可能的替代，默认为-1(无限制)。最后一个参数 count 表示填充执行的次数。该函数的语法格式如下：

```
mixed preg_replace_callback(mixed pattern,callback callback,mixed subject
[,int limit=-1[,int &$count]])
```

在下面实例中自定义一个方法，该方法保存一个自定义数组用来判断拼接，然后在自定义方法外使用 reg_replace_callback()函数来调用自定义方法 translate()进行截取显示，详细代码如下。

```
<?php
function ms($match){
    $arr = array('Welcome' => '欢迎','to' => '光临','window' => '窗内',
    'connection' => '网');
    return $match[0]."(".$arr[$match[0]].")";
}
$string = "Welcome to window connection!";
$result = preg_replace_callback("/\w+/", 'ms', $string);
echo $result;
?>
```

通过以上代码可知，自定义方法 ms()中保存了一个 arr 数组，该数组中保存了英文单词对应的中文翻译，然后读取数组中的数据并拼接起来返回，再使用 preg_replace_callback()方法进行匹配，第一个参数是匹配正则，ms 是调用自定义函数，执行结果为：

```
Welcome(欢迎) to(光临) window(窗内) connection(网)!
```

7. preg_split()函数

preg_split()函数与 split()函数相似，用来拆分给定的正则表达式的字符串。该函数的语法形式如下：

```
array preg_split(string pattern, string string [, int limit [, int flags]])
```

该函数有 4 个参数：pattern 表示要搜索的模式，值为一个字符串；string 表示输入的字符串，limit 表示如果指；最后一个参数 flags 用于控制函数的返回内容，该参数的可选值有如下 3 种。

- PREG_SPLIT_NO_EMPTY：如果这个标识被设置，只有非空值将返回 preg_split()。

- PREG_SPLIT_DELIM_CAPTURE：如果选用此值，括号中的表达模式分隔符将被捕获并返回。
- PREG_SPLIT_OFFSET_CAPTURE：如果设定本标识，对每个出现的匹配结果也同时返回其附属的字符串偏移量。注意这改变了返回的数组的值，使其中的每个单元也是一个数组，其中第一项为匹配字符串，第二项为其在 subject 中的偏移量。

为了方便大家理解，下面通过一个简单的拆分字符串小示例来说明如何使用该函数，代码如下所示：

```
<?php
$str = 'duanshaozhi';
$chars = preg_split('//',$str,-1,PREG_SPLIT_NO_EMPTY);
print_r($chars);
?>
```

上述代码中，首先创建变量 str 保存需要拆分的字符串，接下来则是调用 preg_split()函数，执行结果如下所示：

```
Array ( [0] => d [1] => u [2] => a [3] => n [4] => s [5] => h [6] => a [7] =>
o [8] => z [9] => h [10] => i )
```

5.3.5　实例描述

通过以上基础知识的学习，想必大家一定对使用 PHP 中的正则表达式有了一定的了解，那么本节将使用以上学习到的基础知识来完成用户登录验证的实例，进一步加强大家对 PHP 中的正则表达式的应用。该实例仅有一个简单的用户登录功能，当用户提交登录信息时，通过使用 PHP 中的正则表达式做一些对数据的基本验证等功能。

5.3.6　实例应用

【例 5-2】 用户登录验证

登录验证是每个网站登录时必不可少的一项操作，定义前台登录页面表单内容的代码如下所示：

```
<FORM language=JavaScript name=FrontPage_Form1 action=login.php method=post
target=_top>
<TD style="TABLE-LAYOUT: fixed; FONT-SIZE: 9pt; COLOR: #000000; WORD-BREAK:
break-all; FONT-FAMILY: 宋体,MingLiU, Arial"
        align=middle width=114>邮箱账号
</TD>
       <TD style="TABLE-LAYOUT: fixed; FONT-SIZE: 9pt; COLOR: #000000; WORD-BREAK:
break-all; FONT-FAMILY: 宋体,MingLiU, Arial"
width=205> <INPUT
style="FONT-SIZE: 9pt; COLOR: #000000; FONT-FAMILY: 宋体,MingLiU, Arial"
maxLength=20 size=24 name=user></TD></TR>
        <TR>
```

```
        <TD  style="TABLE-LAYOUT:  fixed;  FONT-SIZE:  9pt;  COLOR:  #000000;
WORD-BREAK:  break-all;  FONT-FAMILY:  宋 体 ,MingLiU,  Arial"  align=middle
width=114>密   码</TD>
        <TD  style="TABLE-LAYOUT:  fixed;  FONT-SIZE:  9pt;  COLOR:  #000000;
WORD-BREAK:    break-all;    FONT-FAMILY:    宋    体    ,MingLiU,    Arial"
width=205> <INPUT style= "FONT-SIZE: 9pt; COLOR: #000000; FONT-FAMILY: 宋
体,MingLiU, Arial" type=password maxLength=20 size=24 name=pass></TD></TR>
        <TR>
</FORM>
```

以上代码设计了登录页面，在该页面中设置了邮箱账号和登录密码，当用户进行提交时，Form 转向 login.php 类进行业务处理，代码如下所示：

```
<?php
$username=$_POST["user"];
$userpwd=$_POST["pass"];
$date=date("Y-m-d H:i");
if(!empty($username)){
    $mail="/^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$
/";
    if(!preg_match($mail,$username)){
        echo '请输入正确的邮箱账号！';
    }else
    {
    if (preg_match_all ("/(\d{4})-(\d{1,2})-(\d{1,2}) (\d{1,2}):(\d{1,2})/",
$date, $regs)) {
        echo '欢迎用户'.$username."登录成功，当前系统时间：";
    echo $regs[1][0]."年".$regs[2][0]."月".$regs[3][0]."日 ".$regs[4][0]."时
".$regs[5][0]."分";
    } else {
    echo "不正确的时间格式:".$date;
    }
    }
}else{
    echo '请输入邮箱账号！';
}
?>
```

以上代码中，首先通过使用$_POST 来获取用户提交的用户名和密码，然后使用 date()函数获取当前的系统时间，使用 empty()函数判断当前用户名是否为空，定义邮箱正则表达式，使用 preg_match()函数来匹配邮箱是否合法，如果通过验证则使用 preg_match_all()函数进行判断获取当前系统时间是否正确，使用数组拼接格式化当前时间，最后提示用户登录成功。

5.3.7 运行结果

以上实例运行后，显示如图 5-2 所示的登录页面，登录成功后显示的页面如图 5-3 所示。

图 5-2　登录页面

登录成功后，转向提示页面，提示当前登录的用户名和当前系统时间，该系统时间是格式化后的结果，如图 5-3 所示。

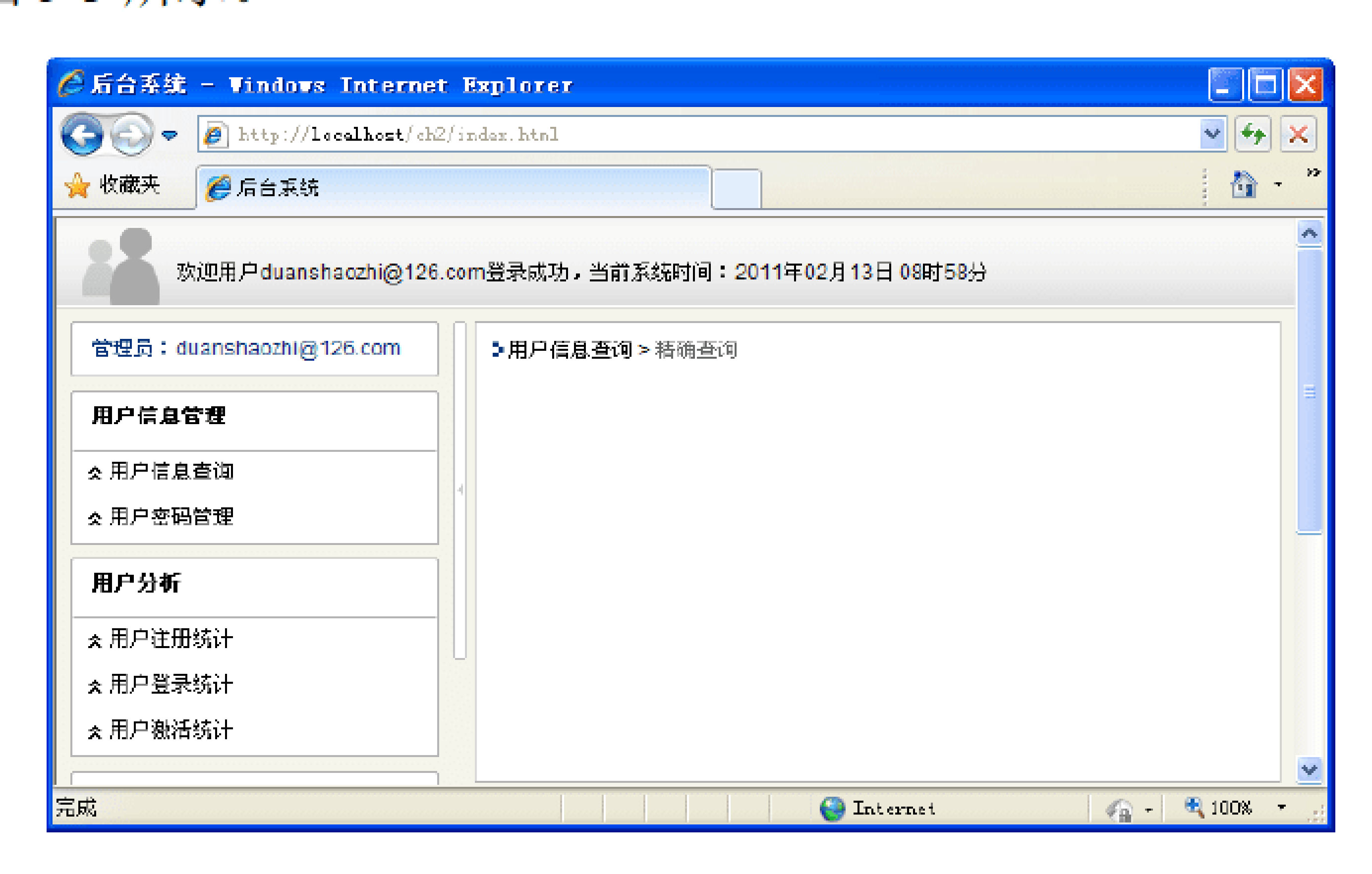


图 5-3　登录成功页面

5.3.8　实例分析



源码解析：

通过以上实例，可以了解使用正则表达式可以针对某些数据进行特定的验证匹配，例如对邮箱、电话号码以及身份证号等有规则的数据进行匹配，同时使用 PHP 内置的一些函数进行判断。

5.4　常见问题解答

5.4.1　何时用 PHP 正则表达式定界符



PHP 正则表达式定界符什么时候用？

网络课堂：http://bbs.itzcn.com/thread-13734-1-1.html

正则表达式一般是写在“//”定界符之间的，但为什么下面的例子没有写定界符？什么时候写，什么时候不用写呢？

```
<?php
if (ereg ("([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $date, $regs)) {
    echo "$regs[3].$regs[2].$regs[1]";
} else {
    echo "Invalid date format: $date";
}
?>
```

【解决办法】：ereg()函数处理的是 POSIX 正则表达式，不需要定界符。preg 系列的函数处理的是 Perl 正则表达式，需要定界符。

并且定界符可以自己选择，只要前后是一对就行。例如我们一般使用“/”符号，但是如果字符串本身有“/”需要匹配，这时还使用“/”作定界符可能就不方便，此时可以使用其他的定界符，如“|”。

5.4.2 PHP 正则表达式替换问题



PHP 正则表达式替换问题？

网络课堂：http://bbs.itzcn.com/thread-13739-1-1.html

在 PHP 中，用正则表达式对下面的文字进行处理。

```
电脑/网络 -> 编程：<img src="logo-zhidao.gif" border="0" alt="百度知道" width="137"
height="46" vspace="3">电脑/网络 -> 硬件电脑/网络 -> 硬件
```

对于上面这段代码，我要将其中的图片源代码去掉该如何操作呢？即无论其中图片的参数如何改变，都能将起始于“<img”和结束于它后面第一个“>”两者之间的内容都去掉，包括这两个符号。

【解决办法】：可以通过使用 preg_replace()函数对内容进行替换，详细代码如下所示：

```
$str='电脑/网络 -> 编程：<img src="logo-zhidao.gif" border="0" alt="百度知道"
width="137" height="46" vspace="3">电脑/网络 -> 硬件电脑/网络 -> 硬件';
echo preg_replace("/<img[^>]+>/", "", $str);
```

代码中第一个参数是需要替换的正则表达式，第二个参数是需要替换成的内容，最后一个参数则是替换的正文。

5.4.3 如何写正则表达式



不会写正则表达式怎么办？

网络课堂：http://bbs.itzcn.com/thread-13741-1-1.html

我的 HTML 文件里面有许多<div>标签，这些标签的属性都相同，只是标签里面的内容不同，例如下面这两个<div>标签：

```
<div style="width:150px;height:30px;color:red;">内容 1 内容 1 内容 1</div>
<DIV STYLE="width:150px;height:30px;color:red;">内容 2 内容 2 内容 2</DIV>
```

上面这两个标签中 style 属性的值是一模一样的，不过标签对之间的内容不一样。现在我要做的是把符合这种样式的<div>标签中的内容全部统一，例如统一为“系统正处于维护阶段，数据暂时无法更新！”。

这里面最让我头疼的地方是 HTML 标签内容不区分大小写，div 也行，DIV 也行，style 也行，STYLE 也行，感觉正则表达式太麻烦了，更何况里面还有那么多特殊字符。而且我上面只是举的一个例子，实际上<div>标签的属性比这还要复杂。

【解决办法】：正则表达式实现不区分大小匹配，好像没有什么难度可言。Perl 风格的正则表达式中使用一个修饰符“i”就可以了。

解决你这个问题，可以将一个完整的<div>内容分成 3 部分，第 1 部分：“<div style="width:150px;height:30px;color:red;">”；第 2 部分：“内容 1 内容 1 内容 1”；第 3 部分：“</div>”。接下来要做的就是分别求出与这 3 部分相匹配的正则表达式。

第 1 部分使用 preg_quote()函数，其代码如下：

```
$string1 = '<div style="width:150px;height:30px;color:red;">';
$pattern1 = preg_quote($string1, '#');
echo $pattern1;
```

preg_quote()函数中的第二个参数“#”用于定义 Perl 正则表达式中的定界符，之所以使用“#”，主要是考虑到字符串中含有大量的“/”。

输出结果为：

```
\<div style\="width\:150px;height\:30px;color\:red;"\>
```

第 2 部分是标签中的内容，可以是任意多个任意字符，可以使用以下代码：

```
$pattern2 = ".*";
```

第 3 部分使用 preg_quote()函数，其代码如下：

```
$string3 = '</div>';
$pattern3 = preg_quote($string3, '#');
echo $pattern3;
```

输出结果为：

```
\</div\>
```

求出这 3 部分的正则表达式后，将它们连接起来构成一个 Perl 正则表达式，代码如下：

```
$pattern = '#'.$pattern1.$pattern2.$pattern3.'#i';
echo $pattern;
```

上述代码在正则表达式最后使用了修饰符“i”，这样可以进行不区分大小写的匹配，也就是不用再担心它是 div 还是 DIV 了。

现在，使用 preg_replace()函数对文本进行替换，代码如下：

```
$string = '<div style="width:150px;height:30px;color:red;">内容 1 内容 1 内容
1</div><DIV STYLE="width:150px;height:30px;color:red;">内容 2 内容 2 内容
```

```
2</DIV>';
$result = preg_replace($pattern, '<div style="width:150px;height:30px;color:red;">
系统正处于维护阶段，数据暂时无法更新！</div>', $string);
echo $result;
```

对于上述代码，将文本中符合正则表达式要求的子字符串替换成“<div style="width:150px;height:30px;color:red;">系统正处于维护阶段，数据暂时无法更新！</div>”，现在让我们看看现在的结果：

```
<div style="width:150px;height:30px;color:red;">系统正处于维护阶段，数据暂时无法
更新！</div>
```

本来有两个<div>标签，现在替换之后只剩下一个了，这显然不是我们想要的。问题出在哪呢，就出在第 2 部分的正则表达式上，“.”表示任意字符，所以“.*”自然也就包括字符串“</div>”，所以它将整个文本按一个<div>标签处理了，而将中间的如下部分全部当成了第 2 部分：

```
内容 1 内容 1 内容 1</div><DIV STYLE="width:150px;height:30px;color:red;">内容 2 内
容 2 内容 2
```

因此，我们在编写第 2 部分的正则表达式时，应该要求第 2 部分中不能出现字符串“</div>”，即应该使用如下正则表达式：

```
$pattern2 = "[^(</div>)]*";
```

修改之后，再看现在的结果：

```
<div style="width:150px;height:30px;color:red;">系统正处于维护阶段，数据暂时无法
更新！</div><div style="width:150px;height:30px;color:red;">系统正处于维护阶段，
数据暂时无法更新！</div>
```

至此就算是大功告成了。

5.5 习　　题

一、填空题

(1) 获取一个字符串的长度，在 PHP 程序中是非常普遍的一个操作，通常使用__________获取字符串长度。

(2) 在 PHP 的正则表达式中，为了编程的方便，还可以使用一些预定义的字符范围，称为__________。

(3) PHP 为使用 Perl 兼容的正则表达式搜索字符串提供了 7 个函数：preg_grep()、preg_match()、preg_match_all()、preg_quote()、preg_replace()、__________和 preg_split()函数。

(4) __________函数用来确保一个字符串的字符数总是大于一个指定值。

二、选择题

(1) 以下哪一项不能把字符串$s1 和$s2 组成一个字符？__________

A. $s1 + $s2　　B. "{$s1}{$s2}"

C. $s1.$s2　　D. implode('', array($s1,$s2))

(2) 给定一个用逗号分隔一组值的字符串，以下哪个函数能在仅调用一次的情况下就把每个独立的值放入一个新创建的数组？________

A. strstr()　　B. 不可能只调用一次就完成

C. extract()　　D. explode()

(3) 以下脚本输出________。

```
<?php
echo'Testing '.1+2.'45';
?>
```

A. Testing 1245　　B. Testing 345

C. Testing 1+245　　D. 245

E. 什么都没有

(4) 如果用“+”操作符把一个字符串和一个整型数字相加，结果将怎样？________

A. 解释器输出一个类型错误

B. 字符串将被转换成数字，再与整型数字相加

C. 字符串将被丢弃，只保留整型数字

D. 字符串和整型数字将连接成一个新字符串

(5) 下面哪个函数能不区分大小写对两个字符串进行二进制比较？________

A. strcmp()　　B. stricmp()

C. strcasecmp()　　D. stristr()

三、上机练习

上机练习： 用户登录。

使用 PHP 技术实现登录验证，当用户提交登录信息时在后台验证用户提交登录信息是否合法，同时做出相应的处理，运行结果如图 5-4 所示。



图 5-4 用户登录



第 6 章　处理文件和解析目录

内容摘要：

在 PHP 中，访问系统中的文件是通过文件路径找到文件，然后再获取文件的相关属性信息，进而操作文件。通过使用 PHP 内置函数可以解析文件路径、修改文件和获取目录重要信息等。本章的前半部分主要讲解内置函数，以及如何使用这些函数获取文件和目录相关信息，例如文件位置、大小、修改时间和最后访问时间等。程序开发人员经常需要查看文件夹和文件，以便收集信息和对文件系统进行必要的更改，例如访问文件系统、读取或写入文件、移动或复制文件或浏览文件夹等。本章的中间部分主要讲解如何在 PHP 中实现这些功能，以及实现这些功能所涉及的函数。

多数人认为 Web 只包含网页，而 HTTP 协议实际上可以上传和下载任何类型的文件，如 Microsoft Office 文档、PDF 文件、可执行文件、MEFG 文件、压缩文件及各种其他类型文件。虽然 FTP 是上传或下载文件的标准方式，但通过网页上传文件也逐渐变得流行。本章的最后两节内容主要讲解 PHP 处理文件的上传和下载。

学习目标：

- 了解如何解析目录路径。
- 掌握访问文件/目录属性的函数。
- 灵活运用打开和关闭文件函数。
- 熟练读取与写入文件。
- 掌握 PHP 处理文件上传和下载。

6.1 解析指定文件的目录路径

PHP 语言的应用范围非常广泛，尤其在对文件的操作上可以灵活地运用 PHP 语言来实现需求。在 PHP 中有很多解析目录路径的函数，这些函数可以获取路径中相关的信息，例如文件名称、路径文件末尾的扩展名、目录部分、路径各个部分和绝对路径等。本节主要介绍可以解析指定文件的目录路径的一些函数。



视频教学：光盘\videos\06\文件目录和属性.avi　　长度：12 分钟

6.1.1 基础知识——解析目录路径

将目录路径解析为各个属性通常很实用，本节将介绍可用来完成这些任务的一些函数。

1. 获取文件名函数

basename()函数返回路径中的文件名部分，该函数语法格式如下：

```
string basename ( string path [, string suffix] )
```

由该函数语法格式可知，basename()函数有两个参数。

- Path：该参数是必需的，表示需要检查的路径。
- Suffix：该参数为可选参数，表示文件的扩展名。如果在使用该函数时提供了 suffix 参数，则不会输出这个扩展名。

下面使用该函数创建一个示例，代码如下：

```
<fieldset>
    <legend>使用 basename()函数获取文件名的示例</legend>
    <?php
            $path="/php/index.php"; //显示带有文件扩展名的文件名
            echo "带有文件扩展名".basename($path)."<br/>";//显示不带有文件扩展名的文件名
            echo "不带有文件扩展名".basename($path,".php");
     ?>
</fieldset>
```

打开 IE 浏览器，浏览 basename()函数的使用效果如图 6-1 所示。

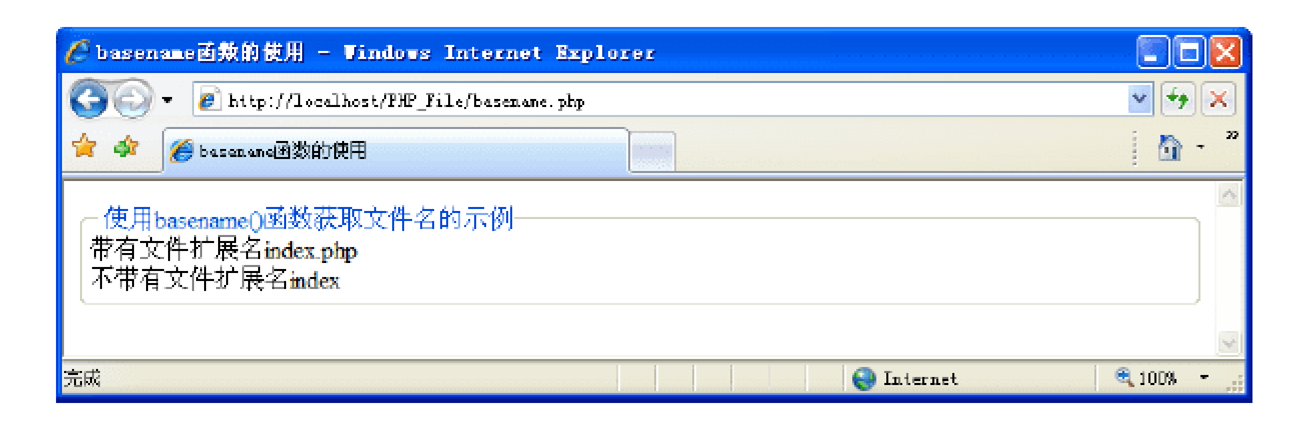


图 6-1　basename()函数的使用效果图

2. 获取目录部分函数

dirname()函数返回路径中的目录部分，该函数的语法格式如下：

```
string dirname ( string path )
```

dirname()函数有一个 path 参数，该参数是必需的，表示要检查的路径，它是指向一个文件的全路径的字符串，该函数返回去掉文件名后的目录名。下面创建一个使用该函数获取路径 path 的目录部分示例，代码如下：

```
<fieldset>
    <legend>使用 dirname 函数的示例</legend>
    <?php
            $path="D:/Program Files/Apache Software Foundation/Apache2.2/
htdocs/PHP_File/php/
index.php";
            echo "<table><tr><td align=\"right\">
            去掉文件名后的目录名：</td><td>".dirname($path)."</td></tr></table>";
    ?>
</fieldset>
```

在浏览器中浏览 dirname()函数的使用效果，如图 6-2 所示。

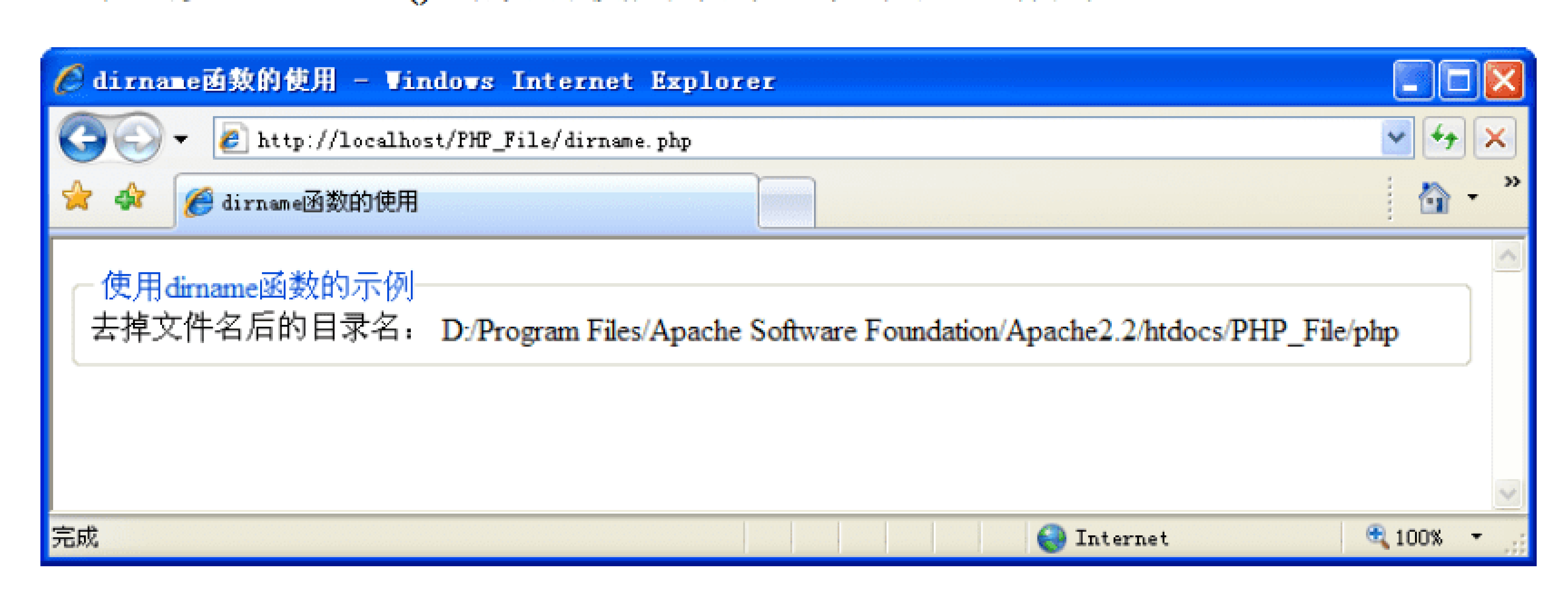


图 6-2　dirname()函数的使用效果图

3. 获取路径中各个部分函数

pathinfo()函数以数组的形式返回文件路径信息，该函数语法格式如下：

```
array pathinfo ( string path [, int options] )
```

pathinfo()函数的参数说明如表 6-1 所示。

表 6-1　pathinfo()函数的参数说明

参　数	是否必需	说　明
path	必需	规定要检查的路径字符串
options	可选	规定要返回的数组元素，默认是 all。可能的值： PATHINFO_DIRNAME　只返回 dirname PATHINFO_BASENAME　只返回 basename PATHINFO_EXTENSION　只返回 extension

该函数返回一个关联数组，其中包括 path 指定的路径中的 3 个部分，目录名、文件名和扩展名，分别通过数组键 dirname、basename 和 extension 来引用。



如果不要求取得所有单元，则 pathinfo()函数返回字符串。

下面创建一个使用 pathinfo()函数的示例获取路径中各个部分，代码如下：

```
<fieldset>
    <legend>使用 pathinfo()函数获取路径中各个部分的示例</legend>
<?php
    $path="web/php/index.php";
    $pathinfo=pathinfo($path);
    echo "获取路径各个部分：<br/>";
    print_r($pathinfo);
    echo "<br/>分解数组后输出：<br/>";
    echo "目录名：".$pathinfo['dirname']."<br/>";
    echo "文件名：".$pathinfo['basename']."<br/>";
    echo "扩展名：".$pathinfo['extension']."<br/>";
?>
</fieldset>
```

打开 IE 浏览器，在浏览器中浏览页面效果，出现如图 6-3 所示的页面。

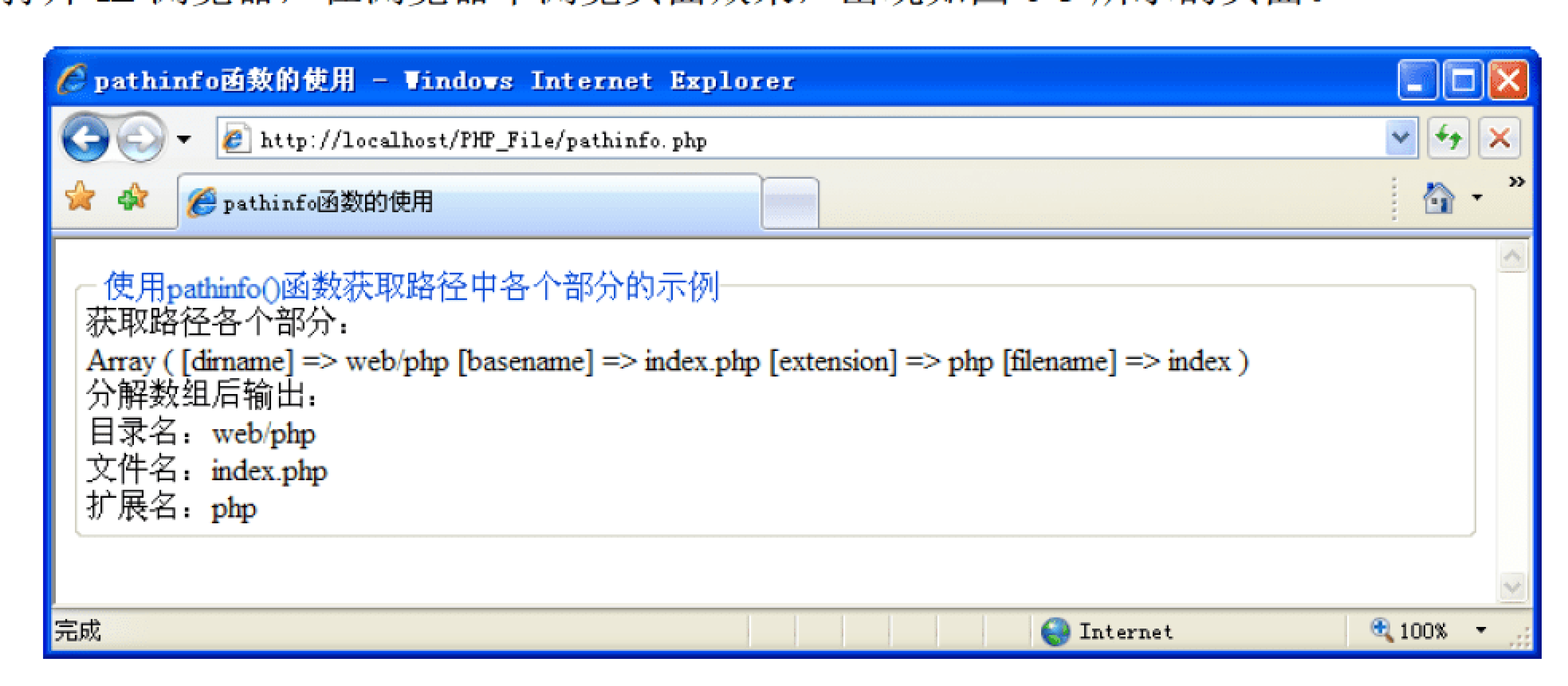


图 6-3　使用 pathinfo()函数的效果图

4．获取规范化的绝对路径名

realpath()函数返回规范化的绝对路径名，该函数的语法格式如下：

```
string realpath ( string path )
```

realpath()函数扩展所有的符号连接并处理输入的 path 中的“'/./', '/../'”以及多余的“'/'”，并返回规范化后的绝对路径名。返回的路径中没有符号连接“'/./'”或“'/../'”成分。该函数执行失败，返回 false，例如文件不存在的情况下。



在 BSD 系统上，如果仅是 path 不存在，PHP 并不会像其他系统那样返回 false。

例如在“D:\Program Files\Apache Software Foundation\Apache2.2\htdocs\PHP_File”目录下有一个文本文件 1.txt，可以使用 realpath()函数解析所有本地路径引用，具体代码如下：

```
<?php
    $path="1.txt";
    echo realpath($path);
?>
```

在浏览器中查看运行效果，在页面中输出结果如下：

```
D:\Program Files\Apache Software Foundation\Apache2.2\htdocs\PHP_File\1.txt
```

6.1.2　实例描述

假设在“D:\Program Files\Apache Software Foundation\Apache2.2\htdocs\PHP_File\web\php”目录下有一个 readMe.txt 文本文件，现在需要获取 readMe.txt 文件的所有相关信息，包括文件名称、文件的目录、文件的扩展名和绝对路径等属性，这些文件的属性可以通过上面的几个函数获取到，从而解析出指定文件的目录路径信息。

6.1.3　实例应用

【例 6-1】 解析指定文件的目录路径

(1) 新建 PHP 文件，命名为 file_1.php。

(2) 在页面中指定所要解析的文件为“web/php/readMe.txt”，并使用 basename()函数获取文件名；使用 pathinfo()函数获取目录名和扩展名；使用 realpath()函数获取文件的绝对路径。代码如下：

```
<?php
    $path="web/php/readMe.txt";
    $suffix=".txt";
    $pathinfo=pathinfo($path);
    echo "你指定的文件名称为：".basename($path,$suffix)."<br/>";
    echo "你指定的文件的目录名为：".$pathinfo['dirname']."<br/>";
    echo "你指定的文件的扩展名为：".$pathinfo['extension']."<br/>";
    echo "你指定的文件的绝对路径为：".realpath($path);
?>
```

6.1.4　运行结果

运行 file_1.php 页面，在页面中输出“web/php/readMe.txt”文件的相关信息，如图 6-4 所示。



图 6-4　解析指定文件的目录路径

6.1.5　实例分析

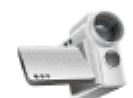


源码解析：

上例中，将“web/php/readMe.txt”和“.txt”两个字符串作为了参数 path 和参数 suffix 传递给了 basename()函数，从而获取到了去掉后缀名的文件名称 readMe，并使用了 dirname()函数获取到了“web/php/readMe.txt”文件的目录名和扩展名。最后使用了 realpath()函数获取到了指定文件的绝对路径，说明使用上面的几个函数可以解析出指定路径下的文件的目录路径信息。

6.2　制作显示用户指定文件的属性信息界面

在 PHP 中有许多可以获取文件属性的函数，例如获取文件类型、大小、最后一次修改时间、最后一次访问时间、文件所属组、文件是否可读和文件是否可写等。本节主要讲解获取文件属性的相关函数。



视频教学：光盘\videos\06\访问文件属性.avi　　　　长度：22 分钟

6.2.1　基础知识——获取文件属性

文件的属性值可以用函数来获取，包括文件类型、文件大小、文件 inode 编号、用户最后一次访问文件时间、最后一次修改时间、上次修改时间、文件相关信息、文件所属组、文件所

有者、文件或目录权限、文件是否可执行及判断文件是否可读和可写，都可以用相应的函数来获取。

1．获取文件类型函数

filetype()函数返回指定文件或目录的类型，该函数的语法格式如下：

```
string filetype ( string filename )
```

filetype()函数有一个 filename 参数，该参数是必需的，规定要检查的文件。如果执行该函数出错则返回 false，如果成功，则返回 7 种可能的值，分别如下。

- fifo：命名管道，常用于将信息从一个进程传递到另一个进程。
- char：字符设备，负责操作系统和设备之间进行无缓冲的数据交换。
- dir：目录。
- block：块设备，例如软件驱动器或 CD-ROM。
- link：符号链接，指向文件指针的指针。
- file：硬链接，作为文件 inode 的指针。只要认为是一个文件，例如文本文档或可执行文件，都返回这个类型。
- unknown：未知类型。



本函数的结果会进入缓存。请使用 clearstatcache()函数来清除缓存。

下面利用一个示例来具体讲解如何使用 filetype()函数获取文件类型，代码如下：

```
<?php
    echo filetype('web/php/readMe.txt')."<br/>";
    echo filetype('web/php/');
?>
```

在浏览器中浏览页面效果，在页面中输出结果如下：

```
file
dir
```

2．获取文件大小函数

filesize()函数返回指定文件的字节数，如果执行成功，则返回文件大小的字节数，否则返回 false，并生成一条 E_WARNING 级的错误。该函数的语法格式如下：

```
int filesize ( string filename )
```

filesize()函数有一个 filename 参数，该参数是必需的，规定要检查的文件。



因为 PHP 的整数类型是有符号的，并且大多数平台使用 32 位整数，filesize()函数在碰到大于 2GB 的文件时可能会返回非预期的结果。对于 2GB 到 4GB 之间的文件通常可以使用“sprintf("%u", filesize($file))”来克服此问题。并且该函数的结果会进入缓存，可使用 clearstatcache()函数来清除缓存。

下面使用该函数创建一个示例，代码见下页。

```
<?php
    $filename = 'web/php/readMe.txt';
    echo $filename . ':  ' . filesize($filename) . ' bytes';
?>
```

在浏览器中运行该段代码，在页面中输出结果为：

```
web/php/readMe.txt: 197 bytes
```

3．获取文件 inode 编号函数

在 PHP 中，使用 fileinode()函数获取文件的 inode 编号。若执行成功，则返回指定文件的 inode 节点号，否则返回 false。该函数的语法格式如下：

```
int fileinode ( string filename )
```



该函数的结果会进入缓存，可使用 clearstatcache()来清除缓存。

下面使用该函数创建一个示例，具体介绍如何使用 fileinode()函数获取文件的 inode 编号，代码如下：

```
<?php
    echo "文件".basename($filename)."的 inode 编号为：".fileinode("web/php/
readMe.txt");
?>
```

在浏览器中浏览效果，在页面中输出结果如下：

```
文件 readMe.txt 的 inode 编号为：0
```

4．获取文件上次被访问的时间函数

在网站开发中，往往需要获取文件上次被访问的时间，此时如果用 PHP 就变得很简单，可以使用 PHP 的内置函数来获取。fileatime()函数返回指定文件上次被访问的时间。如果出错则返回 false，时间以 Unix 时间戳的方式返回。该函数的语法格式如下：

```
int fileatime ( string filename )
```



一个文件的 atime 应该在不论何时读取了此文件中的数据块时被更改。当一个应用程序定期访问大量文件或目录时非常影响性能。有些 Unix 文件系统可以在加载时关闭 atime 的更新以提高这类程序的性能。USENET 新闻组假脱机是一个常见的例子，在这种文件系统下该函数不起作用，并且该函数与上面讲解的几个函数相同，返回的结果都会进入缓存，需要使用 clearstatcache()来清除缓存。

下面创建一个示例，具体介绍如何使用 fileatime()函数来获取文件上次被访问的时间，代码如下：

```
<?php
    $filename = 'web/php/readMe.txt';
    //判断$filename 定义的文件是否存在
    if (file_exists($filename)) {
        echo basename($filename)."文件上次被访问的时间是：" . date("F d Y H:i:s.",
```

```
fileatime($filename));
    }
?>
```

在浏览器中运行该代码所在的页面，输出结果如下：

```
readMe.txt 文件上次被访问的时间是：January 17 2011 16:00:00.
```

5．获取文件上次 inode 修改时间函数

使用 filectime()函数可以获取文件上次 inode 修改时间。该函数返回指定文件上次 inode 被修改的时间，如果出错则返回 false，时间以 Unix 时间戳的方式返回。该函数的语法格式如下：

```
int filectime ( string filename )
```



在大多数 Unix 文件系统中，当一个文件的 inode 数据被改变时，该文件则被认为修改了。某些 Unix 说明文本中把 ctime 说成是该文件建立的时间，这是错误的。在大多数 Unix 文件系统中没有 Unix 文件的建立时间。该函数的结果会进入缓存，更多信息参见 clearstatcache()介绍。

下面创建一个示例，具体介绍如何使用 filectime()函数获取文件上次 inode 修改时间，代码如下：

```
<?php
    $filename = 'web/php/readMe.txt';
    if (file_exists($filename)) {
        echo "文件".basename($filename)."上次 inode 修改时间是：" . date("F d Y
H:i:s.", filectime($filename));
    }
?>
```

在浏览器中浏览使用 filectime()函数的效果，在页面中输出如下：

```
文件 readMe.txt 上次 inode 修改时间是：January 18 2011 01:27:26.
```

6．获取文件上次修改时间函数

filemtime()函数返回文件上次被修改的时间，出错时返回 false。时间以 Unix 时间戳的方式返回，可用于 date()函数。该函数语法格式如下：

```
int filemtime ( string filename )
```

filemtime()函数返回文件中的数据块上次被写入的时间，也就是说，文件的内容上次被修改的时间。下面创建一个示例，具体讲解如何使用该函数获得文件上次被修改的时间，具体代码如下：

```
<?php
    echo "上次被修改的时间是：".date("y 年 m 月 d 日 g:i:sa",filemtime("web/php/
readMe.txt"));
?>
```

在浏览器中执行该段代码，在页面中输出结果如下：

```
上次被修改的时间是：11 年 01 月 18 日 6:09:06am
```

7．获取文件相关信息函数

文件的属性除前面讲到的一些外，还有很多相关的信息，在 PHP 中，使用 stat()函数可以获取由 filename 指定的文件的统计信息。该函数的语法格式如下：

```
array stat(string filename)
```

stat()函数如果执行错误，返回 false，并发出一条警告；若成功，则返回一个包含有文件统计信息的数组，该数组具有的单元如表 6-2 所示，数组下标是从 0 开始的。除了数字索引之外，自 PHP 4.0.6 起还可以通过关联索引来访问。

表 6-2　stat()函数返回数组的单元

数字下标	关联键名(自 PHP 4.0.6)	说　明
0	dev	文件所在的设备号
1	ino	文件的 inode 号
2	mod	文件 inode 保护模式，该值确定指派给文件的访问和修改权限
3	nlink	与该文件关联的硬链接数目
4	uid	文件所有者的用户 ID(UID)
5	gid	文件所有者的组 ID(GID)
6	rdev	设备类型
7	size	文件大小，以字节为单位
8	atime	文件的最后访问时间(Unix 时间戳格式)
9	mtime	上次修改时间(Unix 时间戳格式)
10	ctime	最后一次修改时间(Unix 时间戳格式)
11	blksize	文件系统的块大小，注意 Windows 平台不可用
12	blocks	所占据块的数目



与 stat()函数类似的函数还有 fstat()函数和 lstat()函数，这 3 个函数的用法类似，并且这 3 个函数返回相同的数组，不同的是 fstat()函数作用于打开的文件指针，而不是文件名；stat()函数需要实际的文件名和路径，而不是指针。lstat()函数和 stat()函数类似，所不同的是如果 filename 参数是符号链接，则 lstat()函数返回该符号链接的状态，而不是该符号链接所指向的文件状态被返回。

下面创建一个示例，具体讲解如何使用 stat()函数获取文件的相关信息，如下面的代码所示：

```
<fieldset><legend>获取文件信息</legend>
<?php
    $filename="web/php/readMe.txt";
    //打开 filename 文件，fopen()函数将在后面章节中讲解
    $fn=fopen($filename,"r");
    $array=stat($filename);
    //输出文件的相关信息数组
```

```
    print_r($array);
    //关闭打开的文件，fclose()函数将在下面讲解
    fclose($fn);

?>
</fieldset>
```

在浏览器中查看运行效果，在页面中输出数组中的元素值：

```
Array
(
    [0] => 3
    [1] => 0
    [2] => 33206
    [3] => 1
    [4] => 0
    [5] => 0
    [6] => 3
    [7] => 197
    [8] => 1295280000
    [9] => 1295330946
    [10] => 1295314046
    [11] => -1
    [12] => -1
    [dev] => 3
    [ino] => 0
    [mode] => 33206
    [nlink] => 1
    [uid] => 0
    [gid] => 0
    [rdev] => 3
    [size] => 197
    [atime] => 1295280000
    [mtime] => 1295330946
    [ctime] => 1295314046
    [blksize] => -1
    [blocks] => -1
)
```

8. 获取文件所属组函数

在 PHP 中，使用 filegroup()函数获取指定文件的组 ID，如果出错则返回 false 以及一个 E_WARNING 级别的错误。组 ID 以数字格式返回，用 posix_getgrgid()函数来将其解析为组名。该函数语法格式如下：

```
int filegroup (string filename)
```

下面创建一个示例，具体介绍如何使用 filegroup()函数获取指定文件的组 ID，代码如下：

```
<?php
    echo  "文件".basename($filename)."的组 ID 为： ".filegroup("web/php/
readMe.txt");
?>
```

运行页面，在页面中输出如下结果：

```
文件 readMe.txt 的组 ID 为：0
```

9. 获取文件所有者函数

在 PHP 中，使用 fileowner()函数获取文件所有的用户 ID，如果出错，则返回 false。该函数的语法格式如下：

```
int fileowner ( string filename )
```



用户 ID 以数字格式返回，用 posix_getpwuid()来将其解析为用户名。该函数的结果会进入缓存，需要使用 clearstatcache()来清除缓存。

fileowner()函数的使用与 filegroup()函数的使用相同，这里不再赘述。

10. 获取文件或目录权限

在 PHP 中，使用 fileperms()函数获取文件或目录的权限，如果出错则返回 false。该函数的语法格式如下：

```
int fileperms ( string filename )
```

该函数的结果会进入缓存，需要使用 clearstatcache()函数来清除缓存。下面创建一个 fileperms.php 页面，具体讲解如何使用该函数返回文件或目录的权限，如下面的代码所示：

```
<fieldset>
    <legend>文件权限示例</legend>
   <?php
    $filename="web/php/readMe.txt";
        $filepath="web/php";
        echo "文件权限：".fileperms($filename)."<br/>";
        echo "目录权限：".fileperms($filename)."<br/>";
        echo "以八进制返回文件权限：".substr(sprintf("%o", fileperms($filename)), -4);
    ?>
</fieldset>
```

在浏览器中运行 fileperms.php 页面，在页面中输出文件和目录权限信息，如图 6-5 所示。

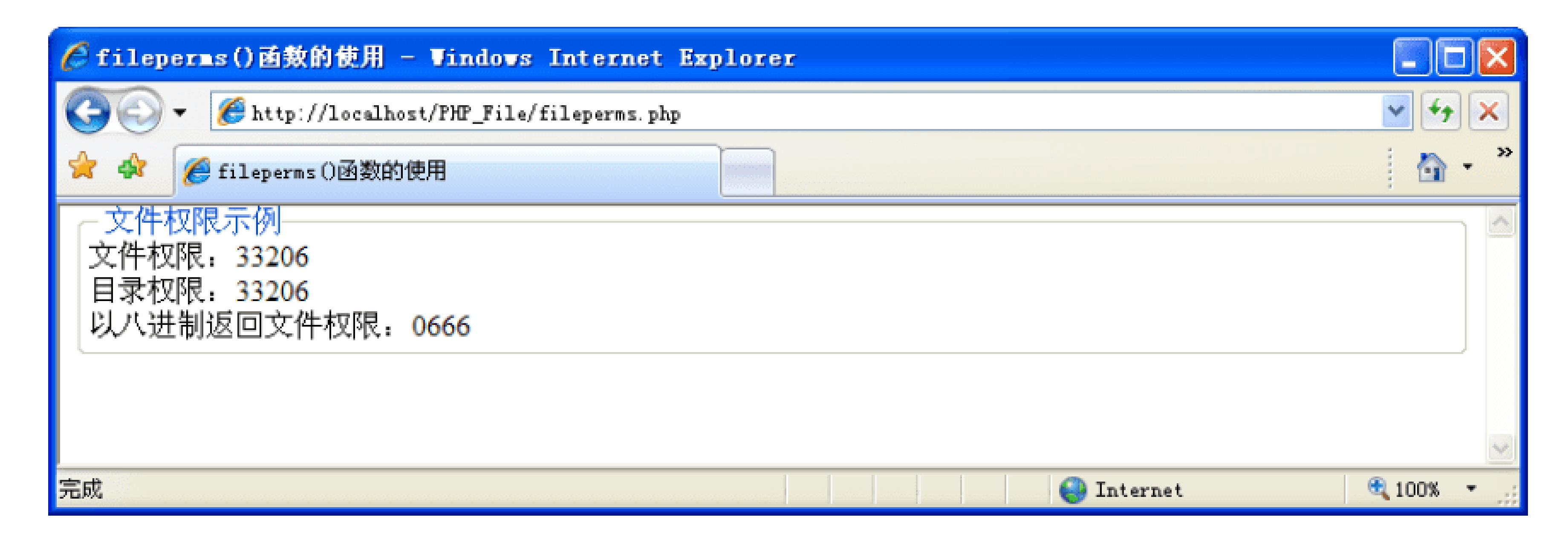


图 6-5　文件的返回权限

11．获取文件是否可执行函数

当访问的文件为可执行文件时，将可能造成系统的崩溃或者其他一些损害，因此判断指定文件是否可执行在 Web 开发中是非常重要的。在 PHP 中，使用 is_executable()函数来判断指定文件 filename 是否可执行，如果文件存在且可执行，则返回 true，否则返回 false。语法格式如下：

```
bool is_executable(string filename)
```



is_executable()从 PHP 5.0.0 版本起可用于 Windows 系统。

下面创建 isexecutable.php 页面，使用 is_executable()函数来判断文件是否可执行，代码如下：

```
<fieldset>
    <legend>判断指定文件是否是可执行文件</legend>
         <?php
            $file="web/php/FirefoxChinaEdition-latest.exe";
            if(is_executable($file))
            {
                echo basename($file)."文件是可执行文件<br/>";
            }
            else
            {
                echo basename($file)."文件不是可执行文件<br/>";
            }
        ?>
</fieldset>
```

在浏览器中运行 isexecutable.php 页面，在页面中输出结果信息，如图 6-6 所示。

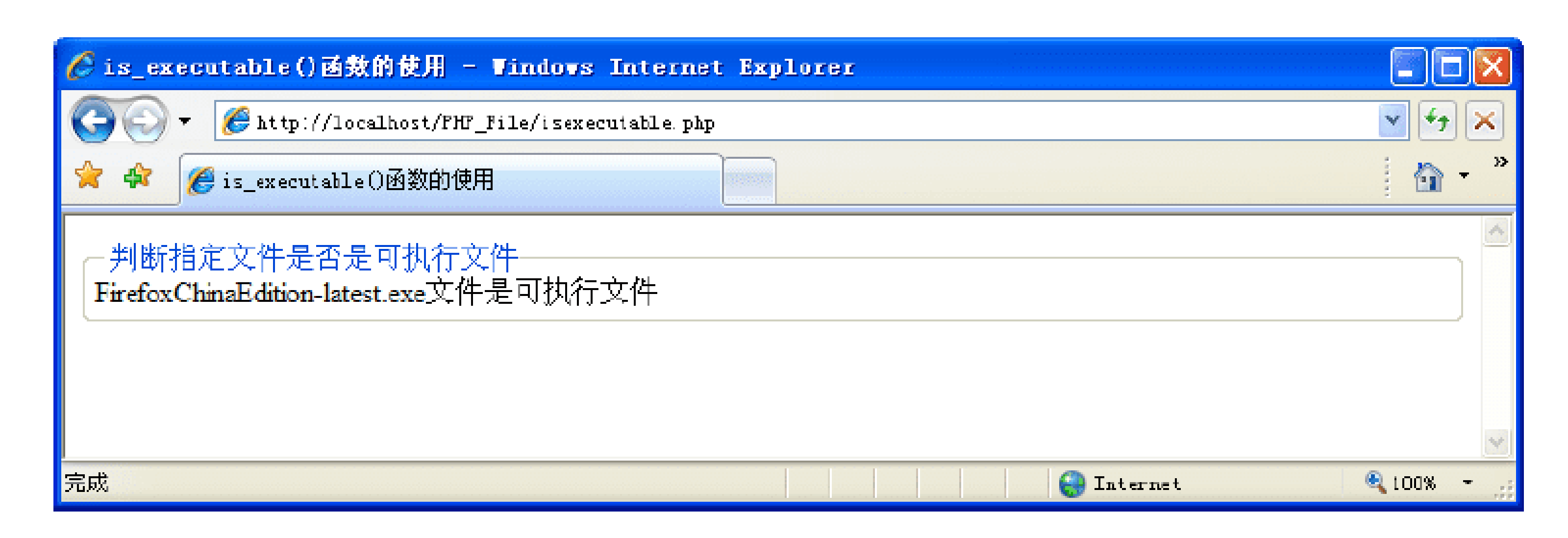


图 6-6　is_executable()函数的使用效果图

12．判断文件是否为可读或者可写函数

读取文件内容显示在页面中或者向已经存在的文件写入内容，这些功能在一个网站中时常会用到。如果某个文件不可读或者不可写的情况下，还能完成读取文件内容和写入文件的功能吗？因此需要在此操作之前判断该文件是否可读或者可写。

在 PHP 中，使用 is_readable()函数判断指定文件是否可读。如果由路径指定的文件或目录存在并且可读，则返回 true，否则返回 false。该函数的语法格式如下：

```
bool is_readable(string filename)
```

is_writable()函数判断指定的文件是否可写。如果由路径指定的文件或目录存在并且可写，则返回 true，否则返回 false。该函数的语法格式如下：

```
bool is_writable(string filename)
```

下面用一个示例来讲解如何使用这两个函数判断指定文件是否可读或者可写。isreadorwrite.php 代码片段如下：

```
<fieldset>
    <legend>判断指定文件是否可读或可写</legend>
<?php
        $filename="web/php/readMe.txt";
        if(is_readable($filename))
        {
            echo "文件".basename($filename)."是可读的<br/>";
        }
        else
        {
            echo "文件".basename($filename)."是不可读取的<br/>";
        }
        if(is_writable($filename))
        {
            echo "文件".basename($filename)."是可写的<br/>";
        }
        else
        {
            echo "文件".basename($filename)."是不可写的";
        }
?>
</fieldset>
```

运行 isreadorwrite.php 页面，在页面中输出判断的结果信息，如图 6-7 所示。

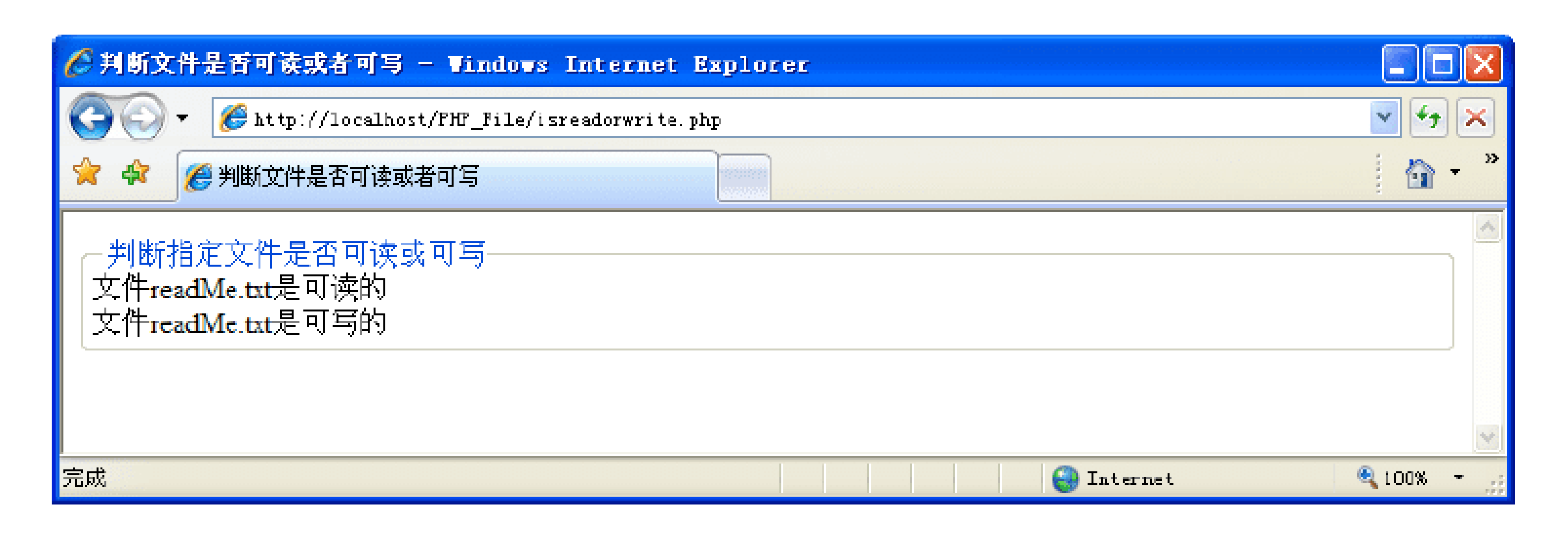


图 6-7 判断指定文件是否可读或可写

6.2.2 实例描述

后台管理系统中常常会有一个【文件管理】模块，这个模块用来管理用户指定的文件，是否要移除或者修改等操作，在对文件进行操作之前，需要让用户了解文件的相关信息，以防用

户误删等。下面我们就来制作显示用户指定文件的相关信息界面。

6.2.3　实例应用

【例 6-2】 制作显示用户指定文件相关信息界面

(1) 在 PHP_File 项目的根目录下新建 right.php 页面，在页面中获取用户指定文件的相关信息，并在页面中输出，关键代码如下：

```
<body>
<?php
    //指定文件路径
    $filename="php/index.php";
    //获取文件名称
    $name=basename($filename);
    //获取文件类型
    $filetype=filetype($filename);
    //获取文件大小
    $filesize=filesize($filename);
    //获取文件的 inode 编号
    $fileinode=fileinode($filename);
    //获取用户最后一次访问时间
    $lasttime=date("m-d-y g:i:sa",fileatime($filename));
    //获取文件最后一次修改时间
    $lastuptime=date("m-d-y g:i:sa",filectime($filename));
    //获取文件所属组
    $filegroup=filegroup($filename);
    //获取文件权限
    $fileperms=fileperms($filename);
    //判断文件是否是可执行函数
    if(is_executable($filename))
    {
        $executresult="是";
    }
    else
    {
        $executresult="否";
    }
    //判断文件是否可读
    if(is_readable($filename))
    {
        $readresult="是";
    }
    else
    {
        $readresult="否";
    }
    //判断文件是否可写
```

```
    if(is_writable($filename))
    {
        $writeresult="是";
    }
    else
    {
        $writeresult="否";
    }

?>
<table class="table" cellspacing="1" cellpadding="2" width="99%" align="center"
border="0">
    <tbody>
        <tr>
            <th class="bg_tr" align="left" colspan="2" height="25">
                <?php echo $name ?>文件信息统计
                </th>
        </tr>
        <tr>
            <td class="td_bg" width="50%" height="23">
                文件类型：<span class="TableRow2"><?php echo $filetype?></span>
            </td>
            <td class="td_bg" width="50%">
                文件大小：<span class="TableRow1"><?php echo $filesize?></span>
            </td>
        </tr>
        <tr>
            <td class="td_bg" width="50%" height="23">
                文件 inode 编号:<span class="TableRow2"><?php echo $fileinode;?>
</span>
            </td>
            <td class="td_bg" width="50%">
                最后一次访问时间：<span class="TableRow1"><?php echo $lasttime;?>
</span>
            </td>
        </tr>
        <tr>
            <td class="td_bg" width="50%" height="23">
                最后一次修改时间:<span class="TableRow2"><?php echo $lastuptime;?>
</span>
            </td>
            <td class="td_bg" width="50%">
                文件所属组：<span class="TableRow1"><font color="#FF0066"><?php
echo $filegroup;?></span>
            </td>
        </tr>
        <tr>
            <td class="td_bg" width="50%" height="23">
                文件权限：<span class="TableRow2"><?php echo $fileperms;?>
```

```
</span>
            </td>
            <td class="td_bg" width="50%">
                是否是可执行文件:<span class="TableRow1"><?php echo $executresult;?>
</span>
            </td>
        </tr>
        <tr>
            <td  class="td_bg">
                文件是否可读：<span class="TableRow1"><?php echo $readresult;?>
</span>              </td>
            <td class="td_bg" width="50%">
                文件是否可写：<span class="TableRow1"><?php echo $writeresult;?>
</span>
            </td>
        </tr>
    </tbody>
</table>
</body>
```

(2) 在 PHP_File 项目的根目录下新建 top.html 页面和 left.html 页面，最后新建 admin_index.html 页面，在最后页面中使用框架将其他三个页面整合，代码如下：

```
<html>
    <head>
        <title>管理中心</title>
        <meta http-equiv="Content-Type" content="text/html;charset=utf-8"> </head>
        <frameset rows="59,*" frameborder="no" border="0" framespacing="0">
            <frame    src="top.html"    noresize="noresize"    frameborder="0"
name="topFrame" marginwidth="0" marginheight="0" scrolling="no">
                <frameset rows="*" cols="195,*" id="frame">
                    <frame          src="left.html"          name="leftFrame"
noresize="noresize"    marginwidth="0"    marginheight="0"    frameborder="0"
scrolling="auto">
                        <frame  src="right.php"  name="main"  marginwidth="0"
marginheight="0" frameborder="0" scrolling="yes">
                </frameset>
        <frame   src="UntitledFrame-1"   noresize="noresize"   frameborder="0"
name="bottomFrame" marginwidth="0" marginheight="0" scrolling="no">
        <noframes>
    <body></body>
        </noframes>
        </frameset>
</html>
```

6.2.4　运行结果

运行 admin_index.html 页面，单击页面左侧的【文件管理】导航菜单，在右侧的框架中显示 right.php 页面，在该页面中输出文件信息，如图 6-8 所示。



图 6-8 显示用户指定文件的相关属性信息

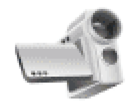
6.2.5 实例分析


源码解析：

上例中，指定了“$filename”变量的值为“php/index.php”，实际在开发中，这个值是传递过来的，可以是用户输入自己想要查找的文件路径或是选择后把相应的值传递到 right.php 页面中，并赋值给“$filename”变量，并不是在页面中指定文件路径，这样是不合理的。

6.3 检测系统盘目录属性信息

导致计算机运行缓慢最主要的原因就是系统盘的空闲容量小，那么在程序开发中，怎么计算出系统盘的空间大小和剩余容量呢？PHP 提供了两个函数可以完成这些功能的实现，分别是 disk_free_space()函数和 disk_total_space()函数，这两个函数不仅能计算出系统盘的空间大小和剩余容量，还能计算出服务器上所有磁盘空间的大小和磁盘空间的剩余容量。本节主要讲解这两个函数以及如何使用 PHP 语言计算目录的大小。


视频教学：光盘\videos\06\文件目录属性.avi　　　　长度：7 分钟

6.3.1 基础知识——访问目录属性信息函数

计算目录和磁盘大小在各种应用程序中都是常见的任务，本节将介绍完成此项任务的标准 PHP 函数。

1. 获取目录所在磁盘可用空间函数

在 PHP 中，使用 disk_free_space()函数返回目录中的可用空间，该函数的语法格式如下：

```
float disk_free_space(string directory)
```

disk_free_space()函数有一个 directory 参数，该参数规定要检查的目录，即它是一个目录的字符串。该函数将根据相应的文件系统或磁盘分区返回可用的字节数。

下面创建一个示例，具体讲解如何使用 disk_free_space()函数获取目录所在磁盘分区的可用空间，代码如下：

```
<?php
    // $df 包含根目录下可用的字节数
    $df = disk_free_space("/");
    //在 Windows 下
    $cfree=disk_free_space("C:");
    $dfree=disk_free_space("D:");
    echo "PHP_File 目录所在磁盘分区的可用空间为：".$df."<br/>";
    echo "系统盘的剩余空间为：".$cfree."<br/>";
    echo "D 盘的剩余空间为：".$dfree."<br/>";
?>
```

在浏览器中执行该段代码，在页面中输出的结果如下：

```
PHP_File 目录所在磁盘分区的可用空间为：9752084480
系统盘的剩余空间为：1868906496
D 盘的剩余空间为：9752084480
```

2. 获取磁盘总容量函数

在 PHP 中，使用 disk_total_space()函数获取一个目录的磁盘总大小，该函数语法格式如下：

```
float disk_total_space(string directory)
```

disk_total_space()函数有一个 directory 参数，该参数是必需的，用来规定要检查的目录。directory 参数是一个目录的字符串。该函数将根据相应的文件系统或磁盘分区返回所有的字节数。



该函数返回的是该目录所在的磁盘分区的总大小，因此在给出同一个磁盘分区的不同目录作为参数所得到的结果完全相同。在 Unix 和 Windows 200x/XP 中都支持将一个磁盘分区加载为一个子目录，这时正确使用该函数就很有意义。

下面创建一个示例，具体介绍如何使用该函数获取一个目录的磁盘总大小。代码如下：

```
<?php
    //返回目录所在磁盘的总大小
    $df = disk_total_space("/");
    //在 Windows 下
    $ctotal=disk_total_space("C:");
    $dctotal=round($ctotal/1048576,2);
    $dtotal=disk_total_space("D:");
    echo "PHP_File 目录所在磁盘的总大小为：".$df."<br/>";
    echo "系统盘的总大小为：".$dctotal."MB<br/>";
```

```
    echo "D 盘的总大小为：".$dtotal."<br/>";
?>
```

在浏览器中查看运行效果，在页面中输出结果如下：

```
PHP_File 目录所在磁盘的总大小为：20577763328
系统盘的总大小为：9811.73MB
D 盘的总大小为：20577763328
```

在该示例中，返回值的数量单位是兆字节(MB)，这是由于在代码中将 disk_total_space()函数返回值除以 1048576(1MB)而得到的。



1TB=1024GB

1GB=1024MB

1MB=1024KB

所以：1MB=1024*1024=1048576KB。

3．获取目录大小的函数

在前面几节中讲到了获取文件大小的函数、获取磁盘剩余空间大小的函数和获取磁盘空间总大小的函数，但还没有介绍如何获取一个指定目录的大小。其实在 PHP 中，并没有提供可以获取目录大小的函数，为此不得不采用另一种解决方法，该方法需要通过 PHP 的一些自定义函数来获取。

下面就做一个实例来获取目录大小，代码如下：

```
<?php
    //计算文件夹大小的函数
    function countDirSize($dir)
    {
        //打开文件
        $handle = opendir($dir);
        //读取目录中的每个文件
        while (false!==($FolderOrFile = readdir($handle)))
        {
            //过滤掉某些未知的文件
            if($FolderOrFile != "." && $FolderOrFile != "..")
            {
                //确定文件大小并进行合并
                if(is_dir("$dir/$FolderOrFile")) {
                    $sizeResult += getDirSize("$dir/$FolderOrFile");
                } else {
                    $sizeResult += filesize("$dir/$FolderOrFile");
                }
            }
        }
        //关闭文件目录
        closedir($handle);
        //返回目录大小
        return $sizeResult;
```

```
    }
    //把字节转换为正常的 K、M 之类的函数：
    function get_real_size($size) {
        $kb = 1024; // Kilobyte
        $mb = 1024 * $kb; // Megabyte
        $gb = 1024 * $mb; // Gigabyte
        $tb = 1024 * $gb; // Terabyte
        if($size < $kb) {
            return $size." B";
        }else if($size < $mb) {
            return round($size/$kb,2)." KB";
        }else if($size < $gb) {
            return round($size/$mb,2)." MB";
        }else if($size < $tb) {
            return round($size/$gb,2)." GB";
        }else {
            return round($size/$tb,2)." TB";
        }
    }
    //要获取的目录
    $mydir="web/php";
    //计算出目录大小
    $size_zip=countDirSize($mydir);
    //转换单位
    $size_zip=get_real_size($size_zip);
    echo $mydir."目录的大小为：".$size_zip;
?>
```

执行上面的代码，在页面中输出如下：

```
web/php 目录的大小为：8.95MB
```

6.3.2　实例描述

一台计算机的系统盘可以调度、监控和维护系统，它负责管理计算机系统中各种独立的硬件，使它们可以协调工作。系统盘可以容纳很多软件，使得计算机使用者可以把系统软件和其他软件将计算机当做一个整体，而不需估计到底每个硬件是如何工作的。因此，对于一台计算机来说，系统盘起着关键的作用，而它的空间容量更是使用者所关心的。如果系统盘的空间本身就小，而系统软件是必须安装在系统盘中才能保证计算机正常运行的，这样就导致了系统盘的可用容量变小，从而，使计算机运行速度变慢，这样不利于计算机使用者。因此，要适时清理计算机系统盘的垃圾文件。下面，编写用来检测系统盘空间容量的代码，以便清晰地看到系统盘的信息。

6.3.3　实例应用

【例 6-3】 检测系统盘目录属性信息

在项目的根目录下新建 file_3.php 页面，在页面中使用 disk_free_space()函数和 disk_total_space()函数获取系统盘(默认 C 盘)的目录属性信息。代码如下：

```
<?php
                $dirve="C:\\";
                //获取用户计算机系统盘可用空间大小
                $freesize=round(disk_free_space($dirve)/1048576,2);
                echo "系统盘的可用空间为：".$freesize."MB<br/>";
                //获取用户计算机系统盘总大小
                $totalsize=round(disk_total_space($dirve)/1048576,2);
                echo "系统盘的总大小为：".$totalsize."MB<br/>";

?>
```

6.3.4 运行结果

运行 file_3.php 页面，在页面中输出系统盘的可用空间和总空间大小信息，如图 6-9 所示。



图 6-9 检测系统盘目录信息

6.3.5 实例分析



源码解析：

上例中，使用 disk_free_space()函数获取了“C:\\”目录下的属性信息，包括可用空间大小和总大小，这样就计算出了“已用空间”大小，让用户很清晰地看到自己计算机的系统盘是否该清理垃圾以加快运行速度。

6.4　打开和关闭文件函数

文件是记录在外部介质中的数据集合，因此它在程序开发中是非常关键的。用户经常需要访问文件并读取数据等，而在访问文件之前文件是关闭的，该文件是不能访问的。因此首先要打开文件，然后才能进行读取、写入、复制和粘贴等操作，最后还需要关闭文件。本节主要讲解在 PHP 中打开和关闭文件的两个标准函数。

 **视频教学：**光盘\videos\06\打开和关闭文件.avi　　　　长度：9 分钟

在处理文件内容之前，通常需要建立与文件资源的连接。而且结束该资源的操作之后，应当关闭连接。

1．打开文件函数

在 PHP 中，使用 fopen()函数可以打开文件或者 URL，如果打开失败则返回 false。该函数的语法格式如下：

```
resource fopen ( string filename, string mode [, bool use_include_path [, resource
context]] )
```

fopen()函数有 4 个参数。

- filename：该参数是必需的，规定要打开的文件或者 URL。
- mode：该参数是必需的，规定要求到该文件/流的访问类型。
- use_include_path：该参数是可选的，如果需要在 use_include_path 中检索文件的话，可以将该参数设为 1 或 True。
- context：该参数是可选的，规定文件句柄的环境。context 是可以修改流的行为的一套选项。

fopen()函数中 mode 参数表示的是访问文件/流的类型，主要是确定用户访问资源的级别，该参数可能的值如表 6-3 所示。

表 6-3　mode 参数可以指定的模式

mode	说　明
“r”	只读方式打开，将文件指针指向文件头
“r+”	读写方式打开，将文件指针指向文件头
“w”	写入方式打开，将文件指针指向文件头，并将文件大小截为零。如果文件不存在，则尝试创建
“w+”	读写方式打开，将文件指针指向文件头，并将文件大小截为零。如果文件不存在，则尝试创建
“a”	写入方式打开，将文件指针指向文件末尾。如果文件不存在，则尝试创建一个新文件
“a+”	读写方式打开，将文件指针指向文件末尾。如果文件不存在，则尝试创建一个新文件

续表

mode	说　明
“x”	创建并以写入方式打开，将文件指针指向文件头。如果文件已存在，则 fopen()函数调用失败，返回 false，并生成一条 E_WARNING 级别的错误信息。如果文件不存在，则尝试创建，这和给定的 open(2)系统调用指定 O_EXCL|O_CREAT 标记是等价的。此选项被 PHP 4.3.2 以及以后的版本所支持，仅能用于本地文件
“x+”	创建并以读写方式打开，将文件指针指向文件头。如果文件已存在，则 fopen()函数调用失败后，返回 false，并生成一条 E_WARNING 级别的错误信息。如果文件不存在，则尝试创建，这和给定的 open(2)系统调用指定 O_EXCL|O_CREAT 标记是等价的。此选项被 PHP 4.3.2 以及以后的版本所支持，仅能用于本地文件

fopen()函数将 filename 指定的名字资源绑定到一个流上。如果 filename 是“scheme://...”的格式，则被当成一个 URL，PHP 将搜索协议处理器(也被称为封装协议)来处理此模式。如果该协议尚未注册封装协议，PHP 将发出一条消息来帮助检查脚本中潜在的问题并将 filename 当成一个普通的文件名继续执行下去。

如果 PHP 认为 filename 指定的是一个本地文件，将尝试在该文件上打开一个流。该文件必须是 PHP 可以访问的，因此需要确认文件访问权限允许访问。如果激活了安全模式或者 open_basedir 则会应用进一步限制。

如果 PHP 认为 filename 指定的是一个已注册的协议，而该协议被注册为一个网络 URL，PHP 将检查并确认 allow_url_fopen 已被激活。如果已关闭，PHP 将发出一个警告，则 fopen 的调用失败。



如果资源是位于本地的文件，PHP 则认为可以使用本地路径或相对路径来访问此资源，或者可以将 fopen()的 use_include_path 参数设置为 1，这样就使 PHP 考虑配置指令 include_path 中指定的路径。

下面创建一个打开文件的实例，代码如下所示：

```
<?php
    //打开一个位于本地服务器的文本文件的只读流
    $handle=fopen("web/php/readMe.txt","r");
    //打开一个位于本地服务器文本文件写入流
    $ handle =fopen("web/php/readMe.txt","w");
    //打开一个远程页面的只读流
    $ handle =fopen("http://www.itzcn.com","r");
?>
```

在用服务器模块版本的 PHP 时，如果在打开和写入文件上遇到问题，记住要确保所使用的文件是服务器进程所能够访问的。在 Windows 平台上，要小心转义文件路径中的每个反斜线，或者用斜线。

```
<?php
    $handle = fopen("c:\\data\\info.txt", "r");
?>
```

2. 关闭文件函数

在 PHP 中，可以使用 fclose()函数关闭已经打开的文件指针，该函数的语法格式如下：

```
bool fclose (resource handle)
```

参数 handle 是必需的，表示将 handle 指向的文件关闭，如果成功则返回 true，否则返回 false。参数 handle 必须有效，并且是通过 fopen()函数或者 fsockopen()函数成功打开的文件指针。下面创建一个实例，介绍如何使用 fclose()函数关闭文件指针，具体代码如下：

```
<?php
    $handle=fopen("web/php/readMe.txt","r");
    fclose($handle);
?>
```

6.5　制作迎春祝福语展示页面

文件用于存储一些外界的数据，而在 Web 开发中，数据又起着关键性的作用，没有数据无法谈程序。程序员需要操作文件，例如读取文件中的内容显示在页面中。PHP 提供了一些可以读取文件内容的函数，这些函数不仅可以一次读取一个字符，而且还可以一次读取整个文件。本节将介绍 PHP 中读取文件的内置函数。

 **视频教学：**光盘\videos\06\读取文件.avi　　　　长度：18 分钟

6.5.1　基础知识——读取文件

前面一节介绍了如何打开一个文件和关闭文件，这节将介绍打开文件之后如何读取文件内容，及读取完毕之后关闭文件。

1. file()函数

在 PHP 中使用 file()函数可以将整个文件读入一个数组中，该函数的语法格式如下：

```
array file(string filename[, int use_include_path [, resource context]])
```

该函数允许有 3 个参数。

- filename 是必需的，规定要读取的文件路径。
- use_include_path 参数可选。如果需要在 use_include_path 路径中搜寻文件的话，可以将参数设为“1”。
- context 参数可选。规定文件句柄的环境。context 是一套可以修改流的行为的选项，若使用 null，则忽略。

file()函数返回的是一个数组，数组中的每个单元都是文件中相应的一行，包括换行符在内。如果失败 file()返回 False。如果需要在 include_path 中搜寻文件的话，可以将可选参数 use_include_path 设为“1”。



file()函数返回的数组中每一行都包括了行结束符，因此如果不需要行结束符时还需要使用 rtrim()函数。如果 PHP 在读取文件时不能识别 Macintosh 文件的行结束符，可以激活 auto_detect_line_endings 运行时配置选项。

下面创建一个实例，具体讲解在 PHP 中如何读取系统中的文件。具体代码如下：

```
<?php
    // 将一个文件读入数组。本例中通过 HTTP 从 URL 中取得 HTML 源文件
    $lines = file('http://www.example.com/');
    // 在数组中循环，显示 HTML 的源文件并加上行号
    foreach ($lines as $line_num => $line)
    {
        echo "Line #<b>{$line_num}</b> : " . htmlspecialchars($line) . "<br
/>\n";
    }
    // 另一个例子将 Web 页面读入字符串。参见 file_get_contents()
    $html = implode('', file ('http://www.example.com/'));?>
```

保存页面为 file_read.php，并在浏览器中运行该网页，出现如图 6-10 所示的界面。

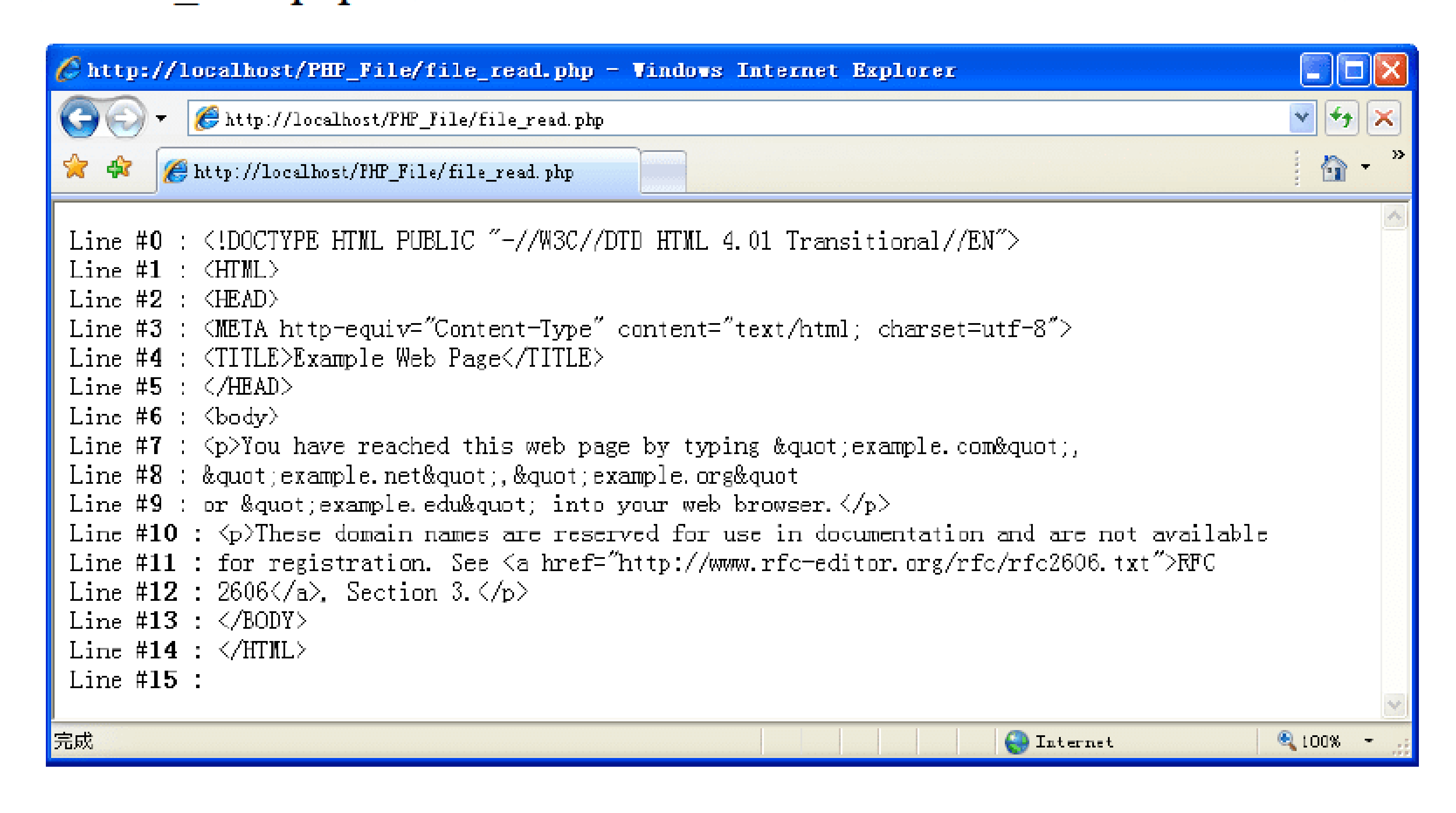


图 6-10　使用 file()函数读取文件

2．file_get_contents()函数

在 PHP 中，使用 file_get_contents()函数可以将整个文件读入一个字符串中，如果失败，将返回 false。该函数语法格式如下：

```
string file_get_contents(string filename [, int use_include_path [, resource
context [, int offset [, int maxlength]]]])
```

file_get_contents()函数将文件的内容读入一个字符串中，如果操作系统支持，还会使用内存映射技术来增强性能。由该函数语法格式可知，该函数有 5 个参数。

- filename：该参数表示要读取的文件名称。
- use_include_path：该参数表示如果需要在 use_include_path 路径中搜寻文件的话，可以将该参数设为“1”。

- context：规定了文件句柄的环境。context 是一套可以修改流的行为的选项，若使用 null，则忽略。
- offset：该参数表示读取文件时开始读取的位置。
- maxlength：该参数表示此次需要读取的字节数。



file_get_contents()函数和 file()类似，不同的是 file_get_contents()函数将文件读入一个字符串中，将在参数 offset 所指定的位置开始读取长度为 maxlen 的内容。file_get_contents()函数是用来将文件的内容读入一个字符串中的首选方法，如果操作系统支持，还会使用内存映射技术来增强性能。

3. fread()函数

在 PHP 中，使用 fread()函数可以读取已经打开的文件。该函数的语法格式如下：

```
string fread ( int handle, int length )
```

fread()函数从文件指针 handle 读取最多 length 个字节。该函数在读取完 length 个字节数，或到达 EOF 时或当一个包可用(对于网络流)时就会停止读取文件，如果出错则返回 false。



fread()函数与其他读取函数不同，使用该函数时，不考虑换行符，因此，只要使用 filesize()函数确定了文件的字符数，就能很方便地读取整个文件。

下面创建一个示例，具体讲解如何使用该函数读取文件，代码如下：

```
<?php
    $filename ="web/php/readMe.txt";
    $handle = fopen($filename, "r");
    $contents = fread($handle, filesize ($filename));
    fclose($handle);
?>
```

4. fgetc()函数

在 PHP 中，使用 fgetc()函数可以从已经打开的文件指针中读取字符，该函数的语法格式如下：

```
string fgetc(resource handle)
```

该函数返回一个包含有一个字符的字符串，该字符从 handle 指向的文件中得到。如果遇到 EOF 则返回 false。文件指针必须有效，并且必须指向一个由 fopen()或 fsockopen()成功打开，但还没有被 fclose()函数关闭的文件。

下面创建一个示例，具体介绍如何使用该函数从已经打开的文件指针中读取字符，具体代码如下：

```
<?php
    $fp = fopen(' php/read.txt', 'r');
    if (!$fp) {
        echo '不能打开文件';
    }
    while (false !== ($char = fgetc($fp))) {
```

```
        echo "$char\n";
    }
?>
```

5. fgets()函数

在 PHP 中，使用 fgets()函数从文件指针中读取一行，出错时返回 false。该函数语法格式如下：

```
string fgets ( int handle [, int length] )
```

该函数从 handle 指向的文件中读取一行并返回长度最多为 length-1 字节的字符串。遇到换行符(包括在返回值中)、EOF 或者已经读取了 length-1 字节后停止，注意先遇到哪一种情况。如果没有指定 length，则默认为 1KB，或者说 1024 字节。下面使用该函数创建一个示例，具体代码如下：

```
<?php
    $handle=fopen("php/re.txt","r");
    if ($handle)
    {
        while (!feof($handle)) {
            $buffer = fgets($handle, 4096);
            echo $buffer;
        }
    }
    fclose($handle);?>
```

在浏览器中查看运行效果，在页面中输出：

```
昔种梁王苑，今移汉将坛。
```

6. fgetss()函数

fgetss()函数从文件指针中读取一行并过滤掉 HTML 标记，该函数的基本语法格式如下：

```
string fgetss ( resource handle [, int length [, string tags]] )
```

fgetss()函数有 3 个参数。

- handle：该参数是必需的，规定要读取的文件。
- length：该参数是可选的，规定要读取的字节数，默认是 1024 字节，该参数在 PHP 5 之前是必需的。
- tags：该参数是可选的，规定将被删除的标签。

该函数与 fgets()函数类似，不同的是 fgetss()函数尝试从读取的文本中去掉任何 HTML 和 PHP 标记。fgetss()函数可以用第三个参数指定哪些标记不被去掉。

下面创建一个 fgetss.php 页面，在该页面中使用 fopen()函数打开上面定义好的 test.html 页面，然后再使用 fgetss()函数读取该文件中的内容，具体代码如下：

```
<?php
    $file = fopen("php/test.html","r");
    echo fgetss($file);
```

```
    fclose($file);
?>
```

在浏览器中浏览该段代码的执行效果，输出 test.html 页面中的内容如下：

```
使用 fgetss()函数的案例页
```

输出的源代码是：

```
<p><b>使用 fgetss()函数的案例页面</b></p>
```

7. fgetcsv()函数

在 PHP 中，使用 fgetcsv()函数从文件指针中读入一行并解析 CSV 字段，该函数的语法格式如下：

```
array fgetcsv(resource handle [, int length [, string speartor [, string enclosure]]])
```

该函数允许有 4 个参数，参数说明如表 6-4 所示。

表 6-4　fgetcsv()函数的参数说明

参　数	是否必需	说　明
handle	必需	规定要检查的文件
length	可选	规定行的最大长度必须大于 CVS 文件内最长的一行。在 PHP 5 中该参数是可选的。在 PHP 5 之前是必需的，如果忽略该参数的话，那么长度就没有限制，不过可能会影响执行效率
separator	可选	设置字段分界符(只允许一个字符)，默认值为逗号(,)
enclosure	可选	设置字段环绕符(只允许一个字符)，默认值为双引号

fgetcsv()函数与 fgets()函数相似，不同的是 fgetcsv()解析读入的行并找出 CSV 格式的字段，然后返回一个包含这些字段的数组。在使用该函数时，如果出错，则返回 false。



CSV 文件中的空行将被返回到一个包含有单个 null 字段的数组，不会被当成错误。该函数对区域设置是敏感的。例如 LANG 设为 en_US.UTF-8，单字节编码的文件就会出现读取错误。如果遇到 PHP 在读取文件时不能识别 Macintosh 文件的行结束符，可以激活 auto_detect_line_endings 运行时配置选项。

下面使用一个示例来具体讲解一下 fgetcsv()函数的应用。创建 fgetcsv.php 页面，在页面中使用 fgetcsv()函数读取 php 文件夹下的 test.csv 文件内容，代码如下：

```
<?php
    $file = fopen("php/test.csv","r");
    print_r(fgetcsv($file));
    fclose($file);
?>
```

在 fgetcsv.php 页面的同级目录下新建 php 文件夹，并在 php 文件夹下新建 test.csv 文件，内容如下：

```
George, John, Thomas, USA
James, Adrew, Martin, USA
```

在浏览器中运行 fgetcsv.php 页面，在页面中输出结果：

```
Array
(
    [0] => George
    [1] => John
    [2] => Thomas
    [3] => USA
)
```

8．fpassthru()函数

在 PHP 中，使用 fpassthru()函数将读取指定文件指针处的所有剩余数据，该函数的基本语法格式如下：

```
int fpassthru ( resource handle )
```

该函数将给定的文件指针从当前的位置读取到 EOF，并把结果写到输出缓冲区。如果发生错误，fpassthru()函数返回 false，否则，fpassthru()函数返回从 handle 读取并传递到输出的字符数目。文件指针必须有效，并且必须指向一个由 fopen()函数或 fsockopen()函数成功打开、但还没有被 fclose()函数关闭的文件。如果已经向文件写入数据，就必须调用 rewind()函数来将文件指针指向文件头。如果既不修改文件，也不在特定位置检索，只想将文件的内容下载到输出缓冲区，应该使用 readfile()函数，这样可以省去 fopen()函数调用。

下面创建一个实例来说明 fpassthru()函数的使用，代码如下：

```
<?php
    $file = fopen("php/test.txt","r");
    // 读取第一行
    fgets($file);
    // 把文件的其余部分发送到输出缓存
    echo fpassthru($file);
    fclose($file);
?>
```

在浏览器中查看运行效果如下所示：

```
湖北・武汉；
广东・广州；
四川・成都；
陕西・西安；
浙江・杭州；68
```

9．readfile()函数

在 PHP 中，使用 readfile()函数读入一个文件并写入输出缓冲区，该函数的语法格式如下：

```
int readfile(string filename [, bool use_include_path [,resource context]])
```

该函数返回从文件中读入的字节数，如果失败，则返回 false，并且除非是以@readfile()函数形式调用，否则会显示错误信息。



如果在 php.ini 文件中“fopen wrappers”已经被激活，则在该函数中可以把 URL 作为文件名来使用。

下面用一个实例来说明 readfile()函数的实用性，具体代码如下：

```
<?php
    echo readfile("php/test.txt");
?>
```

上述代码成功执行后，输出结果如下：

```
河南・郑州；
湖北・武汉；
广东・广州；
四川・成都；
陕西・西安；
浙江・杭州；82
```

6.5.2　实例描述

律回春晖渐，万象始更新。我们告别成绩斐然的 2010，迎来了充满希望的 2011，新春到来，朋友之间、亲人之间、爱人之间、同事之间，都会送上自己的祝福，祝福亲人健康快乐，祝福朋友青春永驻，祝福同事事业有成，等等。

俗话说得好，一份快乐如果分享给 5 个人，就是 5 份快乐。祝福也一样，有时候别人送给自己的祝福，能供其他人分享也是一份莫大的快乐。我们公司最近就做了一个这样的活动，通过系统传递祝福，并把祝福在页面中显示出来。下面就来看看这个功能怎么实现。

6.5.3　实例应用

【例 6-4】 制作迎春祝福语展示页面

用户在页面的表单中输入祝福语后，提交至后台，写入 wishes.xml 文件中，写入后的内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<employees>
<employee>
<name>李赞红</name>
<content>祝您，一帆风顺二龙腾飞三阳开泰四季平安五福临门六六大顺七星高照八方来财九九同心十全十美百事亨通千事吉祥万事如意！祝新年吉祥如意！</content>
</employee>
<employees>
<employee>
<name>张涛</name>
```

```
<content>在这新年里祝愿上帝保佑您！观音菩萨护住您！财神抱住您！爱神射住您！食神吻住您！</content>
<name>王丽丽</name>
<content>晨曦出现的第一缕阳光是我对您的深深祝福，夕阳收起的最后一抹嫣红是我对您的忠心问候。在新年来临之际，送上我真挚祝福：兔年快乐！</content>
<name>马向林</name>
<content>如期而至的不仅仅是节日，还有幸福和快乐。日渐增加的不仅是年龄，还有友情和健康。我的心里一如既往深深祝福:兔年万事如意！</content>
</employee>
</employees>
```

在 file_4.php 页面中使用 PHP 语言读取 wishes.xml 文件中的内容，并在页面中展示出来，其代码如下：

```
<?php
    $parser = xml_parser_create();
        //创建一个 parser 编辑器
    xml_set_element_handler($parser, "startElement", "endElement");
        //设立标签触发时的相应函数。这里分别为 startElement 和 endElenment
    xml_set_character_data_handler($parser, "characterData");
        //设立数据读取时的相应函数
    $xml_file="php/wishes.xml";                //指定所要读取的 xml 文件,可以是 url
    $filehandler = fopen($xml_file, "r");      //打开文件
        //每次取出 4096 个字节进行处理
    while ($data = fread($filehandler, 4096))
    {
        xml_parse($parser, $data, feof($filehandler));
    }
    fclose($filehandler);
    xml_parser_free($parser);                  //关闭和释放 parser 解析器

    $name=false;
    $content=false;
    function startElement($parser_instance, $element_name, $attrs)
        //起始标签事件的函数
    {
        global $name,$content;
        if($element_name=="NAME")
        {
            $name=true;
            $content=false;
        }
        if($element_name=="CONTENT")
            {$name=false;
            $content=true;
        }
    }

    function characterData($parser_instance, $xml_data)  //读取数据时的函数
```

```
    {
        global $name,$content;
        if($content)
            echo    "<font    style='font-size:14px;    font-weight:bold;
color:yellow;'>".$xml_data."</font><br><br>";
        if($name)
            echo    "<font    style='font-size:20px;    font-weight:bold;
color:yellow;'>".$xml_data.": </font><br>";
    }

    function endElement($parser_instance, $element_name) //结束标签事件的函数
    {
        global $name,$content;
        $name=false;
        $content=false;
    }
?>
```

6.5.4　运行结果

在浏览器中运行 file_4.php 页面，效果如图 6-11 所示。

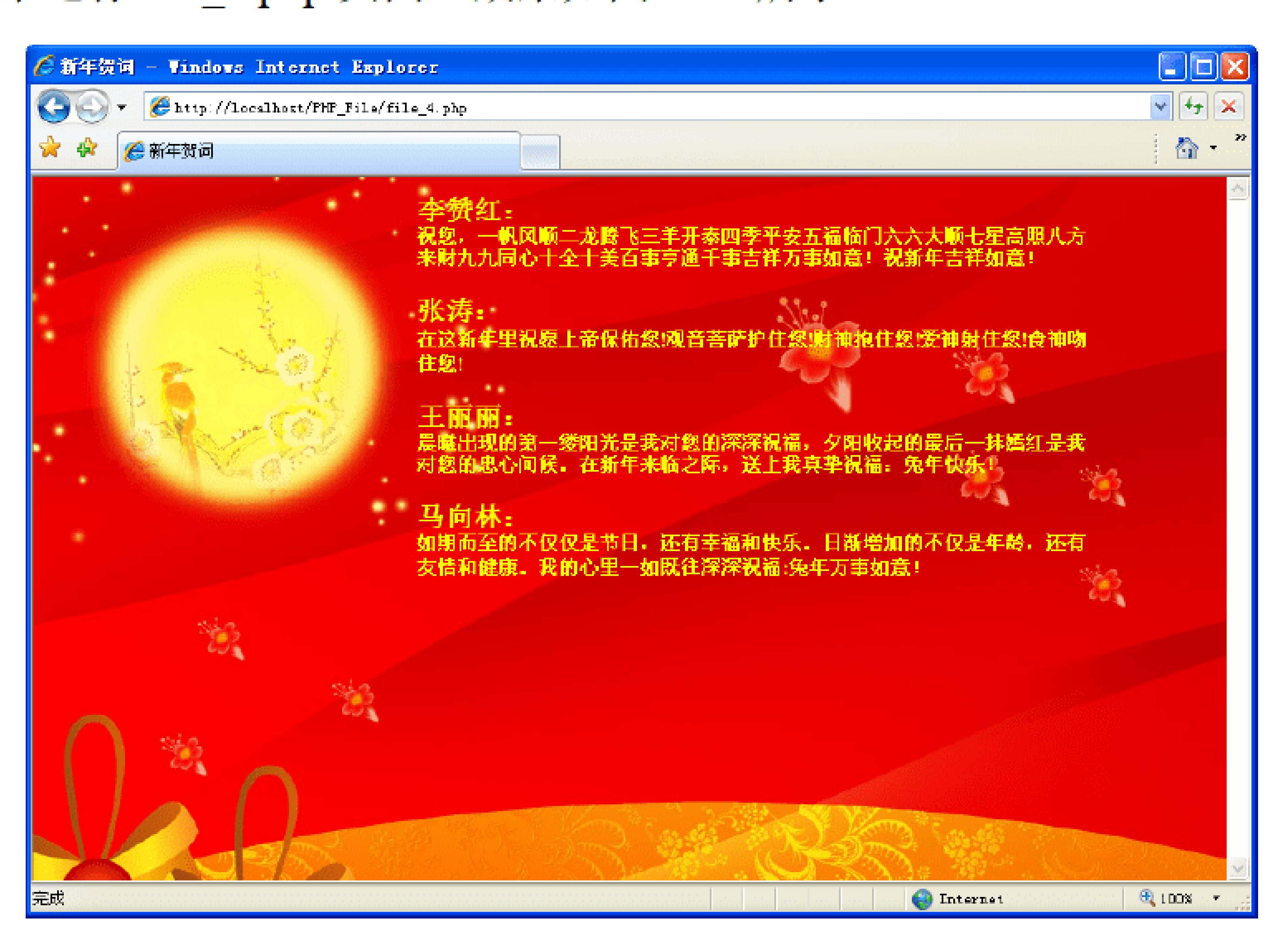
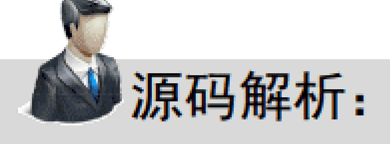


图 6-11　迎春祝福语展示页面

6.5.5　实例分析



源码解析：

上例中，定义了 3 个函数，分别是 startElement()函数、characterData()函数和 endElement()

函数，其中 startElement()函数和 endElement()函数是开始和结束标签事件的函数，使用$name和$content 两个变量来控制是否结束。在 characterData()函数中读取 wishes.xml 文件中的内容，并输出在页面中。建立了这 3 个函数之后，使用 fopen()函数打开文件，循环读取文件中的内容，并显示在页面中。

6.6　制作去掉题目的“咏竹”诗句展示页面

在读取文件或写入文件的时候，需要从指定的位置读取或写入，并非读取全部内容。从文件的开始写入内容，就需要文件指针，即从不同的位置读取以及写入不同的位置。为了实现上述功能，PHP 提供了 3 个函数，分别是 fseek()函数、ftell()函数和 rewind()函数，本节将对这 3 个函数进行详细的介绍。

 **视频教学：**光盘\videos\06\移动文件指针.avi　　长度：6 分钟

6.6.1　基础知识——移动文件指针函数

通常需要在文件中跳转，从不同的位置读取并写入不同的位置，为此，PHP 中提供了 fseek()函数、ftell()函数和 rewind()函数。

1．fseek()函数

在 PHP 中，使用 fseek()函数在打开的文件中定位。该函数的语法格式如下：

```
int fseek(resource handle , int offset [, int whence])
```

该函数允许存在 3 个参数，参数说明如表 6-5 所示。

表 6-5　fseek()函数参数说明

参　数	是否必需	说　明
handle	必需	规定要在其中定位的文件
offset	必需	规定新的位置(从文件头开始以字节数度量)
whence	可选	设置指针位置，可能的值 SEEK_SET：这是默认的值设置指针位置为 offset 字节处，与忽略 whence 效果相同 SEEK_CUR：设置指针位置为当前位置加上 offset 字节 SEEK_END：设置指针位置为文件末尾加上 offset 字节(要移动到文件尾之前的位置，offset 必须是一个负值)

该函数把文件指针从当前位置向前或向后移动到新的位置，新位置从文件头开始以字节数度量。成功则返回 0，规则返回-1。如果没有指定 whence，默认为 SEEK_SET。

下面创建一个实例，具体讲解如何使用 fseek()函数，代码如下：

```
<?php
```

```
    $file = fopen("php/test.txt","r");
    // 读取第一行
    fgets($file);
    // 倒回文件的开头
    fseek($file,0);
?>
```

2．ftell()函数

在 PHP 中，使用 ftell()函数获取打开文件指针读/写的当前位置。该函数的语法格式如下：

```
int ftell(resource handle)
```

该函数返回由 handle 指定的文件指针的位置，也就是文件流中的偏移量。如果出错，则返回 false。文件指针必须是有效的，且必须指向一个通过 fopen()函数或 popen()函数成功打开的文件。在附加模式(加参数“a”打开文件)中 ftell()函数会返回未定义错误。

下面创建一个实例，具体讲解如何运用该函数，代码如下：

```
<?php
    // 打开文件并读/写数据
    $fp = fopen("php/test.txt", "r");
    $data = fgets($fp, 12);
    // 获取当前的文件指针
    echo "当前的文件指针为".ftell($fp); //
    fclose($fp);
?>
```

在浏览器中查看运行效果，在页面中输出：

```
当前的文件指针为 11
```

3．rewind()函数

在 PHP 中，使用 rewind()函数将文件指针倒回到文件的开头位置。该函数的语法格式如下：

```
bool rewind(resource hanle)
```

该函数将 handle 的文件位置指针设为文件流的开头。如果成功则返回 true，失败则返回 false。文件指针必须合法，并且指向由 fopen()函数成功打开的文件。

下面创建一个实例，具体讲解如何使用该函数，代码如下：

```
<?php
    $file = fopen("php/test.txt","r");
    //改变文件指针的位置
    fseek($file,"15");
    //把文件指针设定为 0
    rewind($file);
    echo "当前的文件指针为：".ftell($file);
    fclose($file);
?>
```

在浏览器中执行该段代码，在页面中输出结果如下：

```
当前的文件指针为：0
```

6.6.2 实例描述

竹子清雅隽秀，坚韧挺拔，高风亮节，历来为文人墨客所喜爱。苏东坡云：“宁可食无肉，不可居无竹。”郑板桥咏竹：“咬定青山不放松，立根原在破岩中。千磨万击还坚劲，任尔东西南北风。”可谓千古绝唱！人们赏竹，赞竹，尚竹，不仅是竹的清幽雅致，更有“德”意义。白居易的《养竹记》，就将竹子的“本固”、“性直”、“心空”、“节贞”等特点，比拟君子的品德修养。张九龄的“高洁人相重，虚心世所知。”更是言简意赅。因此，很多公司都在提倡竹子精神，不仅如此，更有甚者在公司的网站展示了“咏竹”这样的首页，来提倡大家发扬竹子精神。

6.6.3 实例应用

【例 6-5】 制作去掉题目的“咏竹”诗句展示页面

首先在 php 文件夹下新建 php6.txt 文件，内容如下。

```
同郭参谋崔仆射淮南节度使厅前竹 刘长卿
昔种梁王苑，今移汉将坛。
蒙笼低冕过，青翠卷帘看。
得地移根远，经霜抱节难。
开花成凤实，嫩笋长渔竿。
霭霭军容静，萧萧郡宇宽。
细音和角暮，疏影上门寒。
湘浦何年变，山阳几处残。
不知轩屏侧，岁晚对袁安。
```

接着在项目的根目录下新建 file_5.php 页面，在页面中使用 fseek()函数给文件定位为 38，即只读取诗歌的内容，而不读取诗歌的题目和作者。在读取文本文件时，使用 fgetcsv()函数来读取的。PHP 代码如下：

```
<?php
    $file=fopen("php/php6.txt","r");
    //文件定位
    fseek($file,"38");
    if(ftell($file)==38)
    {
        $row=1;
        while ($stra=fgetcsv($file,1024,"。")) {
            $num=count($stra);
            $row++;
            for ($i=0;$i<$num;$i++)
            {
                echo    "<font    style='font-size:28px;    font-weight:bold;
font-family:华文行楷'> ".$stra[$i]."</font><br/><br/>";
```

```
            }
        }
        fclose($file);
    }
?>
```

6.6.4　运行结果

运行 file_5.php 页面，在页面中显示读取到的内容，效果如图 6-12 所示。



图 6-12　读取“咏竹”页面

6.6.5　实例分析



源码解析：

在上例中，首先使用 fopen()函数打开了需要读取的文本文件(php/php6.txt)，然后使用 fseek()函数给文件指针定位，接着使用 ftell()函数获取要读取文件指针的当前位置，并判断当前位置是否是诗歌内容的开始(不包含诗歌题目和作者信息)，如果是，使用 fgetcsv()函数则读取文件并输出到页面上。

6.7　制作留言本系统

在上面几节中分别介绍了如何读取文件内容和移动文件指针，本节将介绍如何在打开的文

件中写入内容。为实现这个功能，PHP 提供了两个打开写入文件的函数。

 **视频教学：**光盘\videos\06\写入文件.avi 长度：5 分钟

6.7.1 基础知识——写入文件函数

有时我们需要读取已经存在的文件内容，而有时需要向已经存在的文件中写入内容，本节将为大家讲解向文件写入内容的两个 PHP 函数。

1．fwrite()函数

fwrite()函数的作用主要是写入文件(可安全用于二进制文件)，该函数返回写入文件内容的字节数，语法格式如下：

```
int fwrite(resource handle , string string , [, int length])
```

fwrite()函数允许传递 3 个参数。

- handle：该参数是必需的，规定要写入的打开文件。
- string：该参数也是必需的，规定要写入文件的字符串。
- length：该参数是可选的，规定要写入文件的最大字节数。

fwrite()函数把 string 的内容写入文件指针 handle 处。如果指定了 length，当写入 length 个字节或者写完了 string 以后，写入就会停止。fwrite()函数返回写入的字符数，出现错误时则返回 false。

下面使用一个示例来具体介绍如何使用 fwrite()函数向文件中写入内容。首先创建一个 write.txt 文本文件，文件内容如下：

```
感恩的心感谢有你
我来自偶然
像一颗尘土
……
```

接着创建 write.php 页面，在页面中使用 fwrite()函数向 write.txt 文件中写入内容，代码如下：

```
<?php
    $filename = 'php/write.txt';
    $somecontent = "再加几句歌词：我还有多少爱 我还有多少泪 要苍天知道 我不认输";
    // 首先我们要确定文件存在并且可写
    if (is_writable($filename)) {
   // 在这个例子里，我们将使用添加模式打开$filename
   // 因此，文件指针将会在文件的开头
   // 那就是当我们使用 fwrite()的时候，$somecontent 将要写入的地方
   if (!$handle = fopen($filename, 'a')) {
       echo "不能打开文件 $filename";
       exit;
   }
   // 将$somecontent 写入我们打开的文件中
   if (fwrite($handle, $somecontent) === FALSE) {
      echo "不能写入文件 $filename";
```

```
        exit;
    }
    echo "成功地将 $somecontent 写入文件$filename<br/>";
     echo "写入文件的字节数：".fwrite($handle, $somecontent)."<br/>";
     echo "写入文件的字节数：".fwrite($handle, $somecontent,10);
    fclose($handle);

    } else {
        echo "文件 $filename 不可写";
    }
?>
```

保存并在浏览器中运行 write.php 页面，在页面中输出：

```
成功地将 再加几句歌词：我还有多少爱 我还有多少泪 要苍天知道 我不认输 写入文件
php/write.txt
写入文件的字节数：59
写入文件的字节数：10
```

最后再来看看 write.txt 文件中是否多加了几句歌词，打开 write.txt 文件，内容如下：

```
感恩的心感谢有你
我来自偶然
像一颗尘土
……
再加几句歌词：我还有多少爱 我还有多少泪 要苍天知道 我不认输
```

2. fputs()函数

fputs()函数的功能和 fwrite()函数功能相同，并且用法也基本相同，该函数的语法格式如下：

```
int fputs(resource file , string string , [, int length])
```

该函数是 fwrite()函数的别名，这里不详述。

6.7.2　实例描述

PHP 中提供了读取文件的函数，同时也提供了写入文件的函数，这样，我们可以把文件作为一个“容器”来存储用户输入的信息，并可以从文件中读取存储的信息，这样减轻了数据库很大的 “负担”。比如说，用户留言系统，每天会有成千上万的用户留言，如果用数据库作为存储数据的容器，那么，会给数据库造成很大的压力。因此，可以采用向文件中存储用户的留言信息。每个用户会有一个自己对应的文件，如果有了新的留言，就不需要删除以前该用户的留言内容，直接覆盖就可以了，这样不仅避免了操作数据库表的麻烦，同时也减少了数据库的压力。下面，我们就来做一下留言本系统。

6.7.3　实例应用

【例 6-6】 制作留言本系统

(1) 在 php 文件夹下新建 form.txt 文件，文件内容是空的。

(2) 在项目的根目录下新建 file_6.php 页面，在页面中编辑表单，制作浏览本效果。代码如下：

```
<form action=file_7.php  method=POST>
<font style="font-size:14px; font-weight:bold; color:#000000">请留言:</font>
<textarea name="content" rows=5 cols=40 >
</textarea>
<input name="submit" type="submit" value="留言">
<input type="reset" value="取消">
</form>
```

(3) 在项目的根目录下继续新建 file_7.php 页面，在页面中使用 fwrite()函数向 form.txt 文件中写入用户输入的留言内容。如果用户写入成功，则再次访问 form.txt 文件，读取该文本文件的内容，并在页面中显示出来。file_7.php 页面代码如下：

```
<?php
                if(isset($_POST['submit']))
                {
                    $filename="php/form.txt";
                    $fp=fopen($filename,'at');
                    if(!$fp)
                     {
                         echo"你的信息暂时不能被获取！" ;
                     }
                     else
                     {
                         $textcontent = $_POST['content'];
                         if(!fwrite($fp,$textcontent))
                            echo "您的留言是空的。" ;
                         else
                            echo "您的留言是：".file_get_contents("php/form.txt");
                     }
                }
                else
                    die();
?>
```

6.7.4 运行结果

运行 file_6.php 页面效果如图 6-13 所示。

在页面中输入留言内容，单击【留言】按钮，转到 file_7.php 页面，效果如图 6-14 所示。

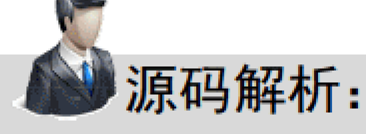


图 6-13　留言本页面效果



图 6-14　从文本文件中读取留言内容

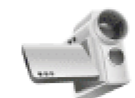
6.7.5　实例分析


源码解析：

上例中，使用 fopen()函数执行打开“php/form.txt”文件，接着判断该文件是否打开，如果已经打开了，将获取到的用户表单中输入的留言信息写入 “php/form.txt”文件中，并判断是否写入成功，如果成功，使用 file_get_contents()函数获取该文本文件中的内容，并在页面中输出。“php/form.txt”文件在此例中充当了容器的作用，显示将用户输入的留言信息保存至该文本文件中，再读取该文件内容，输出到页面中。

6.8 实现目录结构查询功能

前面几节详细介绍了打开文件、读取文件和写入文件的相关函数。实际上，读取目录内容的过程与读取文件非常相似，本节将介绍读取目录内容的相关函数。

 视频教学：光盘\videos\06\读取目录内容.avi　　长度：6 分钟

6.8.1 基础知识——读取目录内容

PHP 提供了 3 个函数用于读取目录内容，分别是 opendir()函数、readdir()函数和 scandir()函数。

1．打开与关闭目录函数

在 PHP 中，使用 opendir()函数打开一个目录句柄，可由 closedir()函数、readdir()函数和 rewinddir()函数调用。若成功，则返回一个目录流，否则返回 false 以及一个 error。可以通过在函数前加上“@”来隐藏 error 的输出。该函数的语法格式如下：

```
resource opendir ( string path [, resource context] )
```

opendir()函数有两个参数。

- path 参数是必需的，规定要打开的目录路径。
- context 参数是可选的，规定目录句柄的环境。context 是可修改目录流的行为的一套选项。



从 PHP 5.0.0 开始，path 参数支持 ftp://URL wrapper。

打开目录以后还需要关闭目录，PHP 提供了一个能完成这项任务的函数——closedir()函数，该函数用来关闭目录句柄。其语法格式如下：

```
void closedir(resource dir_stream)
```

closedir()函数关闭由 dir_stream 指定的目录流，但是目录流必须之前被 opendir()函数所打开。其中，“dir_stream”参数是必需的，规定要关闭的目录句柄。

2．readdir()函数

在 PHP 中，使用 readdir()函数从目录句柄中读取条目。该函数的语法格式如下：

```
string readdir (resource dir_stream)
```

其中，“dir_stream”参数是必需的，规定要使用的目录句柄。该函数返回目录中下一个文件的文件名，文件名以在文件系统中的排序返回。若成功则返回文件名，失败则返回 false。下面创建一个实例，具体介绍如何使用 readdir()函数获取由 opendir()函数打开的目录句柄中的条目，代码见下页。

```
<fieldset>
    <legend>打开并读取目录内容</legend>
    <?php
        if ($handle =opendir('php/')) {
            echo "文件目录：<br/>";
            /* 这是正确地遍历目录方法 */
            while (false != ($file = readdir($handle))) {
                echo "$file<br/>";
            }
            /* 这是错误地遍历目录的方法 */
            while ($file = readdir($handle)) {
                echo "$file<br/>";
            }
            closedir($handle);
        }
    ?>
</fieldset>
```

在浏览器中查看运行效果，如图 6-15 所示。

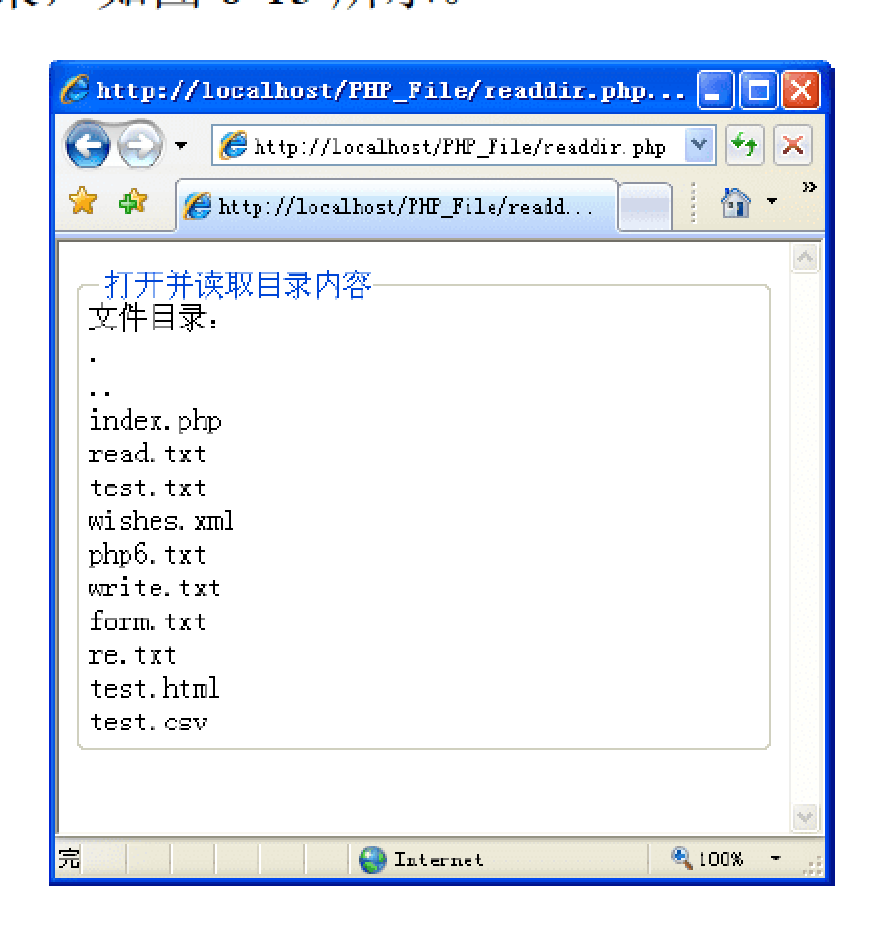


图 6-15　readdir()函数的使用效果图

3．scandir()函数

scandir()函数列出指定路径中的文件和目录。该函数的语法格式如下：

```
array scandir(string directory [, int sorting_order [, resource context]])
```

该函数允许传递 3 个参数，参数说明如表 6-6 所示。

表 6-6　scandir()函数的参数说明

参　数	是否必需	说　明
directory	必需	规定要扫描的目录
sorting_order	可选	规定排列顺序，默认是 0(升序)。如果是 1，则为降序
context	可选	规定目录句柄的环境。context 是可修改目录流的行为的一套选项

该函数返回一个数组，包含有 directory 中的文件和目录，若出错则返回 false。如果 directory 不是目录，则返回布尔值 false，并生成一条 E_WARNING 级的错误。

下面创建一个实例，具体介绍如何使用 scandir()函数，代码如下：

```
<fieldset>
    <legend>打开并读取目录内容</legend>
    <?php
        $dir="php/";
        //默认升序排列
        $files=scandir($dir);
        print_r($files);
    ?>
</fieldset>
```

保存并在浏览器中查看运行效果，如图 6-16 所示。

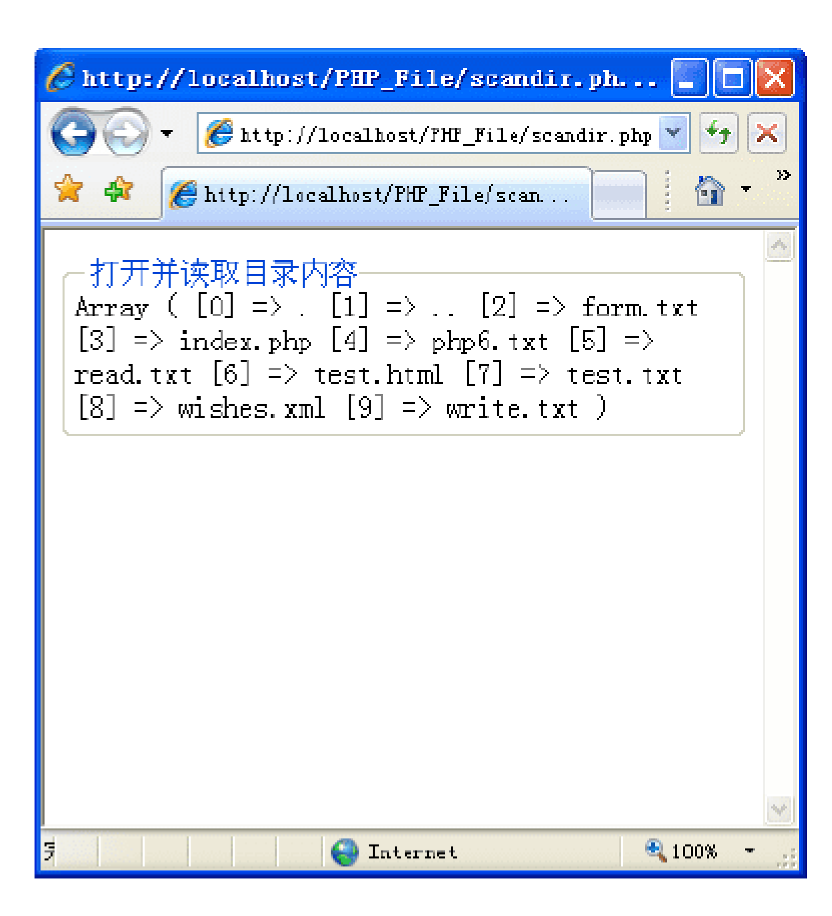


图 6-16　scandir()函数的使用效果图

6.8.2　实例描述

对于一个小型的企业网站，在后台的管理系统中往往会有一个文件和目录管理模块，用来对文件或者目录进行一些操作，在进行操作之前往往也需要把用户要修改的目录或者文件信息查询并显示在网页中，以避免用户操作失误。例如需要修改目录下的几个文件名称或者需要删除该目录下的几个文件，都需要先调查清楚该文件是否已经“没有用途”，没有用就可以删掉。下面，就来实现这个功能，在后台管理系统中查询用户所指定的目录结果，并显示该目录下的所有文件及类型。

6.8.3　实例应用

【例 6-7】 实现目录结构查询功能

(1) 创建 file_8.php 页面，在页面中使用 PHP 语言生成一个表格，输出目录信息。首先编辑函数 viewdir()，并传递一个参数为$ddir，判断传递过去的参数值是否为目录路径，如果是，

打开该目录，循环遍历该目录结构信息，判断该目录下文件的文件类型，最后使用 PHP 语言生成一个表格，把获取到的信息在表格中显示出来。代码如下：

```
<?php
    function viewdir($ddir)
    {
        //判断传递过来的参数值是否是目录结构路径
        if(is_dir($ddir))
        {
            //打开目录结构
            if($handle=opendir($ddir))
            {
                echo "<TABLE cellSpacing=0 cellPadding=2 width='95%' align=center border=0> ";
                //循环遍历目录结构信息
                while(($file=readdir($handle))!=false)
                {
                    //判断文件类型是否为“dir”类型，即文件夹
                    if(filetype($ddir.$file)=='dir')
                    {
                        echo "<TR><TD align=right width=100>文件/目录名称</TD><TD align=left width=200>".$file."</TD><TD align=right width=100>文件类型</TD><TD style='COLOR: #880000'>dir</TD></TR>";
                    }
                    else
                    {
                        //如果是文件，获取文件类型
                        $filetype=filetype($ddir.$file);
                        echo "<TR><TD align=right width=100>文件/目录名称</TD><TD align=left width=200>".$file."</TD><TD align=right width=100>文件类型</TD><TD style='COLOR: #880000'>".$filetype."</TD></TR>";
                    }
                }
                echo "</table>";
            }
        }
    }
    viewdir("./");
?>
```

(2) 编辑 header.htm 页面与 menu.htm 页面。

(3) 新建 index.htm 页面，在页面中使用框架(frameset)，将 header.htm 页面与 menu.htm 页面整合到框架中。

6.8.4　运行结果

运行 index.htm 页面，单击【高级管理】栏目下的【文件管理】，在页面中输出根目录信息，如图 6-17 所示。

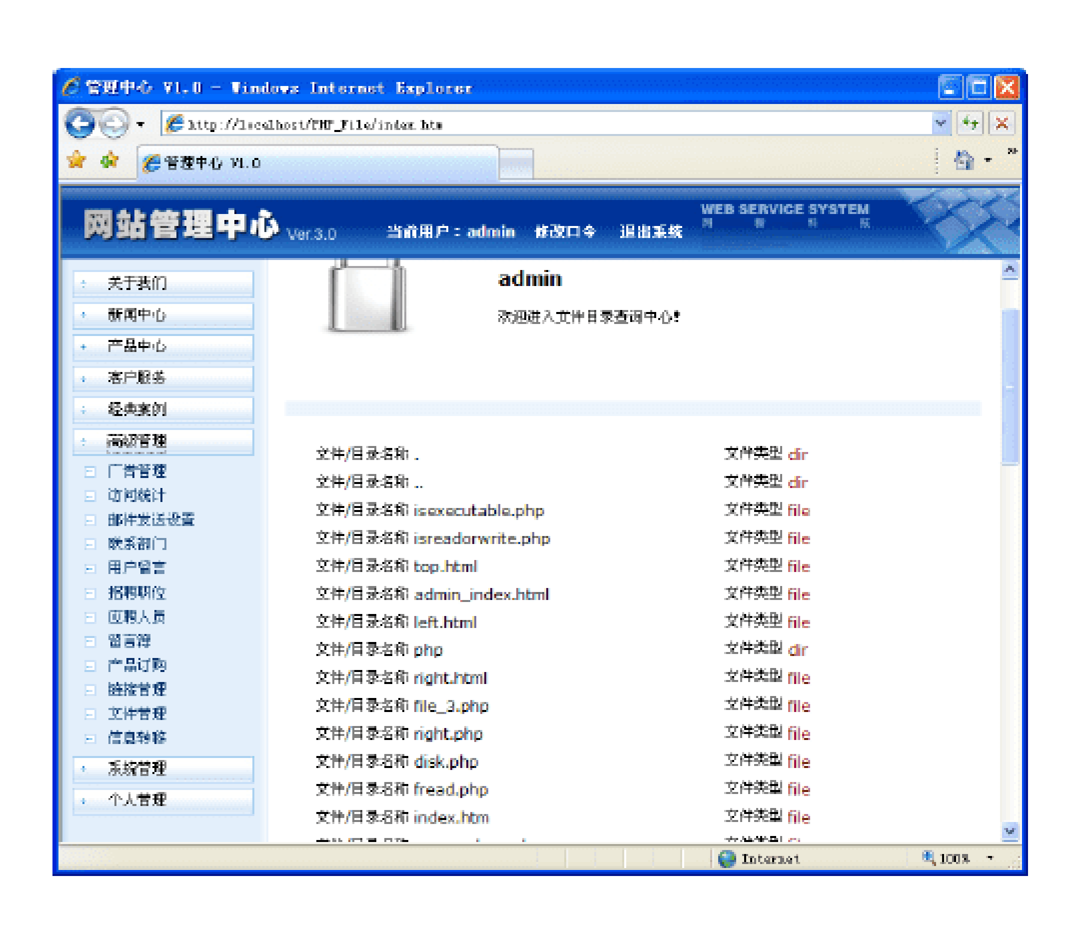


图 6-17 目录文件查询中心页面

6.8.5 实例分析



源码解析：

上例中，定义了一个名为 viewdir()的函数，在函数中判断传递的参数值是否是目录结构，使用 opendir()函数打开该目录结构，接着使用 readdir()函数读取该目录的信息，最后循环遍历该目录结构信息，输出文件/目录名称和文件类型。在这里为了简便，指定的目录路径是固定的，并不是动态的，在做开发时，需要动态传值，把用户选择或者输入的目录路径传递到 viewdir()函数中，而并非直接指定，这是不合理的。

6.9 实现更改头像的功能

通过 PHP 成功地管理文件上传，需要各种配置指令、$_FILES 超级全局变量和编写的 Web 适当表单共同协作。本节将介绍通过 PHP 处理文件上传的一些相关知识。



视频教学：光盘\videos\06\文件上传.avi　　　　长度：5 分钟

6.9.1 基础知识——文件上传

在网页中上传文件已经是 Web 中的一项流行技术，任何一种编程语言都支持上传文件，那么，PHP 也不例外，它需要一些配置指令和函数来完成文件上传功能。

1. PHP 的文件上传/资源指令

有一些配置指令可用于精细调节 PHP 的文件上传功能。这些指令用来确定是否启用 PHP 的文件上传、可允许的最大上传文件大小、可允许的最大脚本内存分配和其他各种重要的资源基准，PHP 的文件上传/资源指令如表 6-7 所示。

表 6-7　PHP 的文件上传/资源指令

指　令	类　型	作 用 域	默认值	说　明
file_uploads	boolean	PHP_INI_SYSTEM	1	该指令确定服务器上的 PHP 脚本是否可以接受文件上传
max_execution_time	integer	PHP_INI_ALL	30	该指令确定 PHP 脚本在注册一个致命错误之前可以执行的最长时间，以秒为单位
memory_limit	integer	PHP_INI_ALL	8M	该指令设置脚本可以分配的最大内存量，以 MB 为单位。该设置中整数值后面必须跟一个 M 才能正常起作用。这可以防止失控的脚本独占服务器内存(甚至在某些条件下是服务器崩溃)。此指令只能在编译时设置了 enable-memory_limit 标志的情况下才生效
upload_max_filesize	integer	PHP_INI_SYSTEM	2M	该指令确定上传文件的最大大小，以 MB 为单位。该指令必须小于 post_max_size 值，因为它只应用于通过 file 输入类型传递的信息，而不应用于通过 POST 实例传递的信息，与 memory_limit 一样，要注意整数值后面有一个 M
upload_tmp_dir	string	PHP_INI_SYSTEM	Null	上传的文件在处理之前必须成功地传输到服务器，所以必须指定一个位置，可以临时放置这些文件，直到文件移到最终目的地为止。这个位置使用该指令来指定
post_max_size	integer	PHP_INI_SYSTEM	8M	该指令确定通过 POST 方法可以接受的信息的最大大小，以 MB 为单位。通常情况下，该指令设置应当大于 upload_max_filesize 值，因为除了上传的文件之外，还可能传递了其他的表单域。与 memory_limit 和 upload_max_filesize 一样，整数值后面必须加上 M

2. $_FILES 数组

$_FILES 超级全局变量很特殊，它是预定义 EGCPFS(Environment、Get、Cookie、Put、Files、Server)超级全局数组中唯一的二维数组。其作用是存储各种与上传文件相关的信息，这些信息

对于通过 PHP 脚本上传到服务器的文件至关重要。



本节介绍的每一项都引用了 file，这只是一个占位符，代表赋给文件上传表单元素的名字。

1) $_FILES['file']['error']

$_FILES['file']['error']数组值提供了与上传文件结果相关的重要信息。总共有 5 个不同的返回值，其中一个表示成功的结果，另外 4 个表示在尝试上传文件中出现的特定错误。返回值的名字和意义如表 6-8 所示。

表 6-8 $_FILES['file']['error']返回值

返回结果	返 回 值	说 明
UPLOAD_ERR_OK	0	没有错误发生，文件上传成功
UPLOAD_ERR_INI_SIZE	1	上传的文件超过了 php.ini 中 upload_max_filesize 选项限制的值
UPLOAD_ERR_FORM_SIZE	2	上传文件的大小超过了 HTML 表单中 MAX_FILE_SIZE 选项指定的值
UPLOAD_ERR_PARTIAL	3	文件只有部分被上传，例如网络错误，导致上传过程中断，就可能发生这种情况
UPLOAD_ERR_NO_FILE	4	用户没有指定上传的文件就提交表单

2) $_FILES['file']['name']

$_FILES['file']['name']数组值指定客户端机器上声明的文件最初的名字，包括扩展名。因此，如果浏览一个名为 vacation.jpg 的文件，并通过表单上传，则此变量的值将是 vacation.jpg。

3) $_FILES['file']['size']

$_FILES['file']['size']数组值指定从客户端上传的文件的大小，以字节为单位。因此，在 vacation.jpg 文件的例子中，此变量可能赋值为 5236，大约为 5KB。

4) $_FILES['file']['tmp_name']

$_FILES['file']['tmp_name']数组值指定上传到服务器后为文件赋予的临时名。这是存储在临时目录(由 PHP 指令 upload_tem_dir 指定)中时所指定的文件名。

5) $_FILES['file']['type']

$_FILES['file']['type']数组值指定从客户端上传的文件的 MIME 类型。因此，在 vacation.jpg 文件的例子中，此变量会赋值为 image/jpeg。如果上传的是 PDF，则赋值为 application/pdf。因为这个数组值有时会得到意外的结果，所以应当在脚本中显示地进行验证。

3. PHP 的文件上传函数

在 PHP 中，提供了两个上传文件的函数，分别是 is_uploaded_file()函数和 move_uploaded_file()函数。本节将介绍这两个函数的使用。

1) is_uploaded_file()函数

在 PHP 中，使用 is_uploaded_file()函数判断指定的文件是否通过 HTTP POST 方法上传。该函数的语法格式如下：

```
bool is_uploaded_file(string filename)
```

如果 filename 所给出的文件是通过 HTTP POST 上传的则返回 true。这可以用来确保恶意用户无法欺骗脚本去访问禁止访问的文件，例如/php/password。上传的文件有可能会对用户或本系统的其他用户显示其内容时，这种检查显得格外重要。为了能使 is_uploaded_file()函数正常工作，字段指定类似于$_FILES['userfile']['tmp_name']的变量，而在从客户端上传的文件名$_FILES['userfile']['name']不能正常运作。下面创建一个实例，来具体讲述 is_uploaded_file()函数的使用。

首先编辑表单页面，代码如下：

```
<form ENCTYPE="multipart/form-data" ACTION="uploadsubmit.php" METHOD="POST">
    <input type="hidden" name="MAX_FILE_SIZE"  value="1000">
    <div align="center"><center> 请选取文件:
    <input NAME="userfile" TYPE="file">
    <input TYPE="submit" VALUE="Send File">
    </center></div>
</form>
```

接着编辑处理表单提交页面——uploadsubmit.php 页面，即处理文件上传页面，代码如下：

```
<?php
if (is_uploaded_file($_FILES['userfile']['tmp_name'])) {
   echo "File ". $_FILES['userfile']['name'] ." uploaded successfully.\n";
   echo "Displaying contents\n";
   readfile($_FILES['userfile']['tmp_name']);
} else {
   echo "Possible file upload attack: ";
   echo "filename '". $_FILES['userfile']['tmp_name'] . "'.";
}
?>
```

在浏览器中运行上传文件页面，效果如图 6-18 所示。上传成功，则跳转至 uploadsubmit.php 页面，效果如图 6-19 所示。



图 6-18　文件上传页面



图 6-19　处理上传文件页面

2)　move_uploaded_file()函数

在 PHP 中，使用 move_uploaded_file()函数将上传的文件移动到新的位置。该函数的语法格式如下：

```
bool move_uploaded_file(string filename,string destination)
```

其中，“filename”参数是必需的，规定要移动的文件；“destination”参数也是必需的，规定文件的新位置。

该函数检查并确保由“filename”指定的文件是合法的上传文件(即通过 PHP 的 HTTP POST 上传机制所上传的)。如果文件合法，则将其移动为由“destination”指定的文件。



如果“filename”不是合法的上传文件，不会出现任何操作，move_uploaded_file()函数将返回 false; 如果“filename”是合法的上传文件，但由于某些原因无法移动，不会出现任何操作，move_uploaded_file()函数将返回 false，此外还会发出一条警告。

使用 move_uploaded_file()函数非常简单，只需要指定所要一定的文件位置及移动后的文件名即可，如下：

```
<?php
    $datetime=date("Ymdhis");
    $newname=$datetime.$_FILES["userfile"]["name"];
    //图片重命名其中 file 是 file 中 name 的属性值
    $path="test/" .$newname;
    //保存路径，这个是保存到本文档的上一级目录的 test 文件夹中
    $a=move_uploaded_file($_FILES['userfile']['tmp_name'],$path);
    echo "移动成功！";
?>
```

6.9.2 实例描述

当用户登录到后台管理系统后，可进入个人中心管理自己的账号、密码、头像等基本信息。在更改头像时，如果用户从未更改过头像，系统将给用户一张默认的头像；如果用户需要更改自己的头像，则需要上传一张 JPEG 格式或者 GIF 格式的图片，上传成功后，在页面中显示新头像。下面，我们就来实现更改头像的功能。

6.9.3 实例应用

【例 6-8】 实现更改头像的功能

(1) 新建 file_9.php 页面，在页面中编辑表单域，代码如下：

```
<form name="form1" method="post" action="" enctype="multipart/form-data">
    上传头像：
    <input type="file" name="file" id="file">
    <input type="submit" name="upload" id="button" value="更改头像...">
</form>
```

(2) 接着，在该页面中编辑处理文件上传的代码，将上传文件的信息以表格的形式展现在页面中。代码片段如下：

```
<?php
    if(isset($_POST["upload"]))                          //单击“更改头像...”按钮
    {
```

```
        if (($_FILES["file"]["type"] == "image/gif")
        //判断文件类型是否为 gif 或者 jpg
        || ($_FILES["file"]["type"] == "image/jpeg")
        || ($_FILES["file"]["type"] == "image/pjpeg"))
        {
            if ($_FILES["file"]["error"] > 0)           //发生错误
            {
                echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
                                                        //输出错误信息
            }
            else
            {
                echo "<TABLE cellSpacing=0 cellPadding=2 width='95%' align=
center border=0><TR><TD align=right width=200>";
                echo " 文 件 名 称 : </TD><TD align=left width=200>" .
$_FILES["file"]["name"] . "</TD></TR>";                 //上传的文件名称
                echo "<TR><TD align=right width=200> 文 件 类 型 : </TD><TD
align=right width=200>" . $_FILES["file"]["type"] . "</TD></TR>";
                                                        //上传的文件类型
                echo "<TR><TD align=right width=200> 文 件 大 小 : </TD><TD
align=right width=200>" . ($_FILES["file"]["size"] / 1024) . " Kb</TD></TR>";
                                                        //文件大小
                echo "<TR><TD align=right width=200>文件临时副本名称: </TD><TD
align=right width=200>" . $_FILES["file"]["tmp_name"] . "</TD></TR>";
                                                        //文件别名
                if (file_exists("uploads/" . $_FILES["file"]["name"]))
                                                        //判断文件是否存在
                {
                    echo "<TR><TD colspan=2>".$_FILES["file"]["name"] . " 已
经存在! </TD></TR> ";
                }                                       //输出存在相同文件提示
                else
                {
                    move_uploaded_file($_FILES["file"]["tmp_name"],
                                                        //开始上传
                    "uploads/" . $_FILES["file"]["name"]);
                    echo "<TR><TD colspan=2> 文 件 存 储 在 : " . "uploads/" .
$_FILES["file"]["name"]."</TD></TR>";                   //输出上传后的路径
                }
                echo "</TABLE>";
            }
        }
        else
        {
            echo "上传文件不存在";                        //输出错误
        }
    }
?>
```

上段代码中“$_FILES["file"]["type"]”中的“file”指的是页面表单域中的文件域“file”。

(3) 在 file_9.php 页面中显示上传成功后的头像信息，代码如下：

```
<?php
        if($_FILES["file"]["name"]=="")
        {
            echo "<img src='1.jpg' width=100 height=100>";
        }
        else
        {
            echo    "<img    src="."uploads/"    .    $_FILES["file"]["name"]."
width=100>";
        }
?>
```

(4) 编辑框架的头部页面(header.htm)和左边菜单页面(menu.htm)。

(5) 新建 file_9_index.htm 页面，在页面中使用框架将 header.htm 页面、menu.htm 页面和 file_9.php 页面整合。

6.9.4 运行结果

运行 file_9_index.htm 页面，第一次显示系统默认头像，如图 6-20 所示。



图 6-20 用户默认头像

选择要上传的头像，单击【更改头像...】按钮，获取上传头像的信息，并显示在页面中，头像也随之更换为新头像，如图 6-21 所示。



图 6-21　头像更新成功

6.9.5　实例分析


源码解析：

在上例中的 file_9.php 页面中，首先使用$_FILES["file"]["type"]数组值来限制用户上传文件的类型只能是 GIF 格式、JPEG 格式或者 JPEG 格式。接着判断上传文件是否合法，如果合法，分别使用$_FILES["file"]["name"]、$_FILES["file"]["type"]、$_FILES["file"]["size"] / 1024、$_FILES["file"]["tmp_name"]获取了上传文件的名称、类型、大小(MB)、临时副本名称。最后实现文件的上传，先判断要上传的文件是否已经存在，如果存在，提示用户已存在，不做上传操作，如果不存在，使用 move_uploaded_file()函数实现了文件的上传功能。

6.10　使用 PHP 实现下载文档功能

很多时候，需要把一个用户上传的文件提供给多方用户下载查看，这种情况下，就需要使用 PHP 下载文件技术。本节将介绍如何使用 PHP 下载文件。


视频教学：光盘\videos\06\文件下载.avi　　长度：3 分钟

6.10.1　基础知识——文件下载

使用 PHP 实现文件下载，首先需要判断用户是否单击了下载链接，如下所示：

```
<a href="?action=download">下载文件到本地</a>
```

接着指定文件的名称和路径，并打开该文件使用函数输入文件类型、大小和内容，代码如下：

```
    header("Content-type:application/octet-stream");
    header("Accept-Ranges:bytes");
    header("Accept-Length:".filesize('1.jpg'));
    header("Content-Disposition:attachment;filename=1.jpg");
```

最后使用 PHP 中的 fread()函数将文件内容直接在页面中输出，让浏览器提示用户下载。所有的这些处理都是在服务器端完成的，因此用户是不会知道文件具体位置信息的，是非常安全的一种下载方法。

6.10.2　实例描述

无论是一个小型公司，还是一个具有很多分公司的大企业，往往需要上传一些文档(会议记录、日程安排、系统说明书等)供领导审核，审核通过之后，部门经理才可以执行相应的命令来完成一件事。而领导在审核时，需要把部门经理上传的文档先下载到本地电脑上查看，下面使用 PHP 来完成下载文档的功能。

6.10.3　实例应用

【例 6-9】 使用 PHP 实现下载文档功能

(1) 编辑 file_10.php 页面，在页面中编辑一个链接，供用户选择是否要下载。代码如下：

```
<a href="?action=download"> 文档下载</a>
```

上面的链接路径是虚拟的，只是用来标识用户是否单击了【文档下载】按钮，即判断用户是否需要下载文档。

(2) 编辑 PHP 代码，处理文件下载。首先要判断用户需要下载的文件是否存在，如果不存在，提示“文件找不到”的信息；如果存在，则打开该文件，指定 Header()函数信息，最后使用 fread()读取文件，关闭打开的文件即可。代码如下：

```
<?php
        if(isset($_GET["action"]))              //是否单击链接
        {
            $file_name = "OA.doc";              //指定文件名
            $file_dir = "web/";                 //指定文件路径
            if (!file_exists($file_dir . $file_name)) {
                                                //检查文件是否存在
                echo "文件找不到";              //输出错误提示
                exit;                           //退出
            }
            else
            {
                $file = fopen($file_dir . $file_name,"r");
                                                //打开文件
```

```
                    Header("Content-type: application/octet-stream");
                                            //输入文件类型
                    Header("Accept-Ranges: bytes");
                   Header("Accept-Length: ".filesize($file_dir . $file_name));
                                            //输入文件大小
                    Header("Content-Disposition:  attachment;  filename="  .
$file_name);                                //输入文件名称
                    echo fread($file,filesize($file_dir . $file_name));
                                            //开始下载
                    fclose($file);          //关闭文件
                    exit;                   //退出
                }
            }
?>
```

(3) 后台管理系统中的页面使用框架，编辑框架的头部页面、左边导航菜单页面和主页面。

(4) 新建 file_10_index.htm 页面，将头部页面、左边导航菜单页面和主页面整合。

6.10.4　运行结果

运行 file_10_index.htm 页面，单击【高级管理】栏目下的【文件管理】，转到文件下载中心页面，在页面中显示文件名称，并存在【文档下载】按钮。当用户单击【文档下载】时，出现如图 6-22 所示的页面效果。

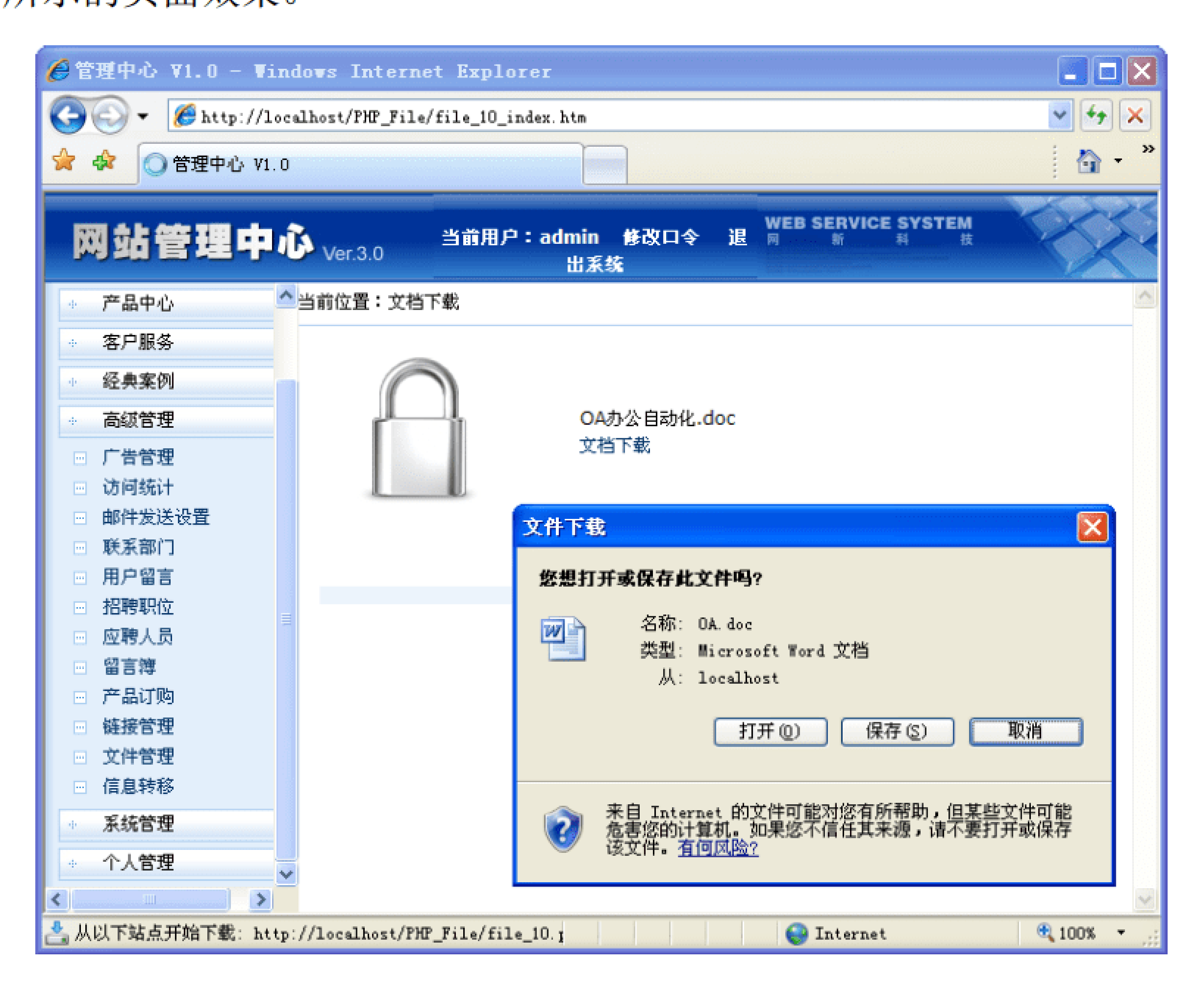


图 6-22　文件下载

单击【保存】按钮，选择保存到本地的路径，即可实现文件的下载功能。

6.10.5 实例分析



源码解析：

在上例中的 file_10.php 页面中，提供了一个超链接供用户选择是否需要下载文档，当用户单击【文档下载】按钮时，触发超链接并传递一个参数名为“action”的参数，接着使用 PHP 中的$_GET()函数判断用户是否需要下载文档。因为超链接时 GET 提交方式，因此使用$_GET(“action”)可获取到“action”参数的值，从而执行文档下载操作。

6.11 常见问题解答

6.11.1 PHP 中文件打开错误的问题



PHP 中文件打开错误的问题。

网络课堂：http://bbs.itzcn.com/thread-12471-1-1.html

执行代码“$file = fopen("/php_kejian/ceshi.txt","w+");”时出现错误，提示如下：

```
Warning: fopen(/php_kejian/ceshi.txt) [function.fopen]: failed to open stream:
No such file or directory in D:\PC_webserver\phproot\php_kejian\Wen_Jian_
Xi_Tong_Han_Shu.php on line 22
```

这是怎么回事？我检查了一下好像没什么问题，为什么会提示这样的错误呢？

【解决办法】：你的文件路径有问题。文件打开时，第 1 个参数不要以“/”开头，改用下面的语句：

```
$file=fopen($_SERVER['DOCUMENT_ROOT']."/php_kejian/ceshi.txt","r+");
```

6.11.2 PHP 读取文件的问题



PHP 读取文件的问题。

网络课堂：http://bbs.itzcn.com/thread-12472-1-1.html

我想实现这样一个功能：有几个关键词(是我按照需要列出来的)，然后，从一个字符串中依次与这几个关键词匹配，找出一样的。

现在的问题是，我想把关键词放在一个文本文件中，或者 CSV 文件中，然后读取。这样做的话，我只要一行一个关键词这样写就可以了吗？

【解决办法】：这个问题很多同学会遇到吧！下面我来告诉大家，可以这样做。

```
$keys=explode(' ',preg_replace("/\s+/",' ',trim(file_get_contents("文件名
"))))
```

然后，把文件内容获取成为一个字符串。再删除前后的空白，把所有的空白替换为一个。

最后按空格分割为数组。这样写这个文本的时候非常自由，可以一行写一个，也可以一行写多个，中间可以用空格、<TAB>等符号分隔，有多余的空行和空格也没关系。

6.11.3　指定格式读取文件



如何指定格式读取文件？

网络课堂：http://bbs.itzcn.com/thread-12473-1-1.html

有一个 date.txt 文件，里面的内容是有格式的，每行一条信息。然后，我用下面的代码读取：

```
$handle = fopen('date.txt','r');    //打开 date.txt 这个文件;
$data = file('date.txt');           //读取全部内容;
print_r ($data);                    //输出
```

输出结果如下：

```
Array (
 [0] => 郑州 安阳 开封 洛阳 信阳
 [1] => 长沙 湘潭 株洲 岳阳 益阳
)
```

现在的问题是：怎么样才能以“" "”这个符号(chr 值为 8)为分隔，将每个城市都加上索引呢？

【解决办法】：PHP 的 file 在读文件之前，无需使用 fopen 打开。可以试试下面的代码：

```
$data = file('date.txt');           //读取全部内容;
foreach ($data as $i=>$line) $data[$i]=explode(chr(8),$data[$i]);
echo '<pre>';
print_r($data);                     //输出
echo '</pre>';
```

6.11.4　打开一个文件替换指定内容后再次写入



如何打开一个文件替换指定内容后再次写入？

网络课堂：http://bbs.itzcn.com/thread-12474-1-1.html

有个文件为 books.html，现在想把 books.html 中的“[我的电脑]”替换成“PHP 学习”，然后再写入 books.html 中。

另外，由于我设置的是半小时执行一次，所以请问有没有什么方法能最大限度节省资源的？

【解决办法】：频繁读取、写入文件时要尽量减少代码的长度。下面是我认为的最节省资源方案，供你参考。

```
<?php
$f='books.html';
file_put_contents($f,str_replace('[我的电脑]','PHP 学习',file_get_contents
($f)));
?>
```

6.12 习　　题

一、填空题

(1) 下面这段程序输出的结果是__________。

```
<?php
    $path="test/readdir.txt";
    $suffix=".php";
    echo basename($path,$suffix);
?>
```

(2) 函数__________用于判断所给文件是否可读，如果由路径指定的文件或目录存在并且可读，则返回 true。

(3) file_get_contents()函数用于将整个文件读入__________中，在页面中的显示不是格式化的。

(4) 如果要读取的文件是规则的，比如每行以“@”符号结尾，并且每行仅出现一个“@”符号，那么可以使用 PHP 中的__________函数读取该文件，解析 CSV 字段。

(5) 下面一段程序中使用了 PHP 中的函数移动了几次指针的位置，那么指针的当前位置为__________。

```
<?php
    $path="test/readdir.txt";
    $file=fopen($path,"r");
    fseek($file,"30");
    rewind($file);
?>
```

二、选择题

(1) 下面一段程序中输出的结果是__________。

```
<?php
    $path="web/php/readMe.txt";
    echo dirname($path);
?>
```

A．web　　B．php
C．web/php　　D．web/php/readMe.txt

(2) 获取用户最后一次访问“readMe.txt”文本文档的时间，并且输出的格式为 December 28 10 16:00:00。下面那段代码是对的？__________

A．

```
<?php
    $path="test/readdir.txt";
    echo date("F d y H:i:s",fileatime($path));
?>
```

B.

```
<?php
    $path="test/readdir.txt";
    echo date("m-d-y g:i:s",fileatime($path));
?>
```

C.

```
<?php
    $path="test/readdir.txt";
    echo date("F d y H:i:s",filemtime($path));
?>
```

D.

```
<?php
    $path="test/readdir.txt";
    echo date("F d y H:i:s",filectime($path));
?>
```

(3) fwrite()函数主要是写入文件，该函数返回__________。

A. 写入的内容　　B. 写入内容的字数

C. 写入内容的字节数　　D. Null

(4) 当需要读取一个“images”文件夹下的目录时，下面哪段代码是正确的？__________

A. readdir(images);

B. readdir("images/");

C. readdir(opendir(images));

D. readdir(opendir("images/"));

三、上机练习

上机练习： 读取 csv 文件中的数据并插入数据库表。

首先在 user.csv 文件中编辑两条数据，每条数据的格式为 username，password。接着在页面中读取编辑好的 user.csv 文件中的内容，并显示数据，如图 6-23 所示。然后在页面中添加超链接，当用户单击【插入数据库表中】按钮时，将读取到的 user.csv 文件中的数据插入数据库 user 的表 table_user 中，并在页面中显示插入结果信息，如图 6-24 所示。



图 6-23　显示读取文件数据信息



图 6-24　将读取到的文件数据插入数据库表中



第 7 章　表单处理与身份认证

内容摘要：

在浏览网页时常常可以看到各式各样的输入表单，有些只是一个文本输入框或提交按钮，有些则比较复杂(如一些免费电子邮箱的注册表单)，除了文本输入框之外，还有单选按钮、复选框、下拉列表框等，虽然复杂程度不同，但表单的组成元素却是相同的。使用 PHP 创建的网站同样可以自动生成这些表单元素进行数据交互。

本章将为大家介绍如何使用 PHP 获取表单数据并对其进行处理，PHP 如何生成表单，PHP、Web 表单和 JavaScript 之间的交互以及 PHP 的身份认证。

学习目标：

- 掌握向 PHP 函数传递表单数据。
- 掌握 PHP 处理多值表单组件。
- 掌握 PHP 自动生成表单。
- 熟悉 PHP 与 JavaScript 集成。

7.1 获取表单元素的值

当一个脚本向另一个脚本传递数据时，GET 方法是默认方法，但是一般希望使用 POST 方法，因为 POST 方法能处理更多的数据。POST 方法发送给 PHP 脚本的提交数据都必须使用$_POST 语法来引用。本节将主要介绍如何使用$_POST 变量来获取表单数据。

 **视频教学：**光盘\videos\07\获取表单元素的值.avi　　长度：10 分钟

7.1.1 实例描述

就像每个家都有一扇门一样，程序也有一道“门”，那就是登录界面。一个用户要想访问程序的关键界面，必须先登录，然后判断该用户是否“合法”，这就相当于主人判断敲门的那个人是否是“善者”一样，如果没有那扇门，可想会带来什么样的后果。那么，下面就用 PHP 来识别一下“门”外的那个人是谁吧！

7.1.2 实例应用

【例 7-1】 获取表单元素的值

首先创建 form_1.php 页面，在页面中定义一个表单，如下面代码所示：

```
<form action="form_1.php" method="post">
        <p>
            用户名：<input type="text" name="username" maxlength="20" />
        </p>
        <p>
            密码：<input type="password" name="userpwd" maxlength="18" />
        </p>
        <input type="submit" name="submit" value="Go! " />
</form>
```

然后，继续在本页面中添加 PHP 代码，使用$_POST 变量来获取表单中的元素值，PHP 代码如下：

```
<?php
        if(isset($_POST['submit'])){
                echo "Hi! ".$_POST['username']."! <br/>";
                echo "你的密码是：".$_POST['userpwd']."! ";
        }
?>
```

7.1.3 运行结果

当用户在表单的两个元素中输入相应的值后，单击 Go！按钮，出现如图 7-1 所示的界面。

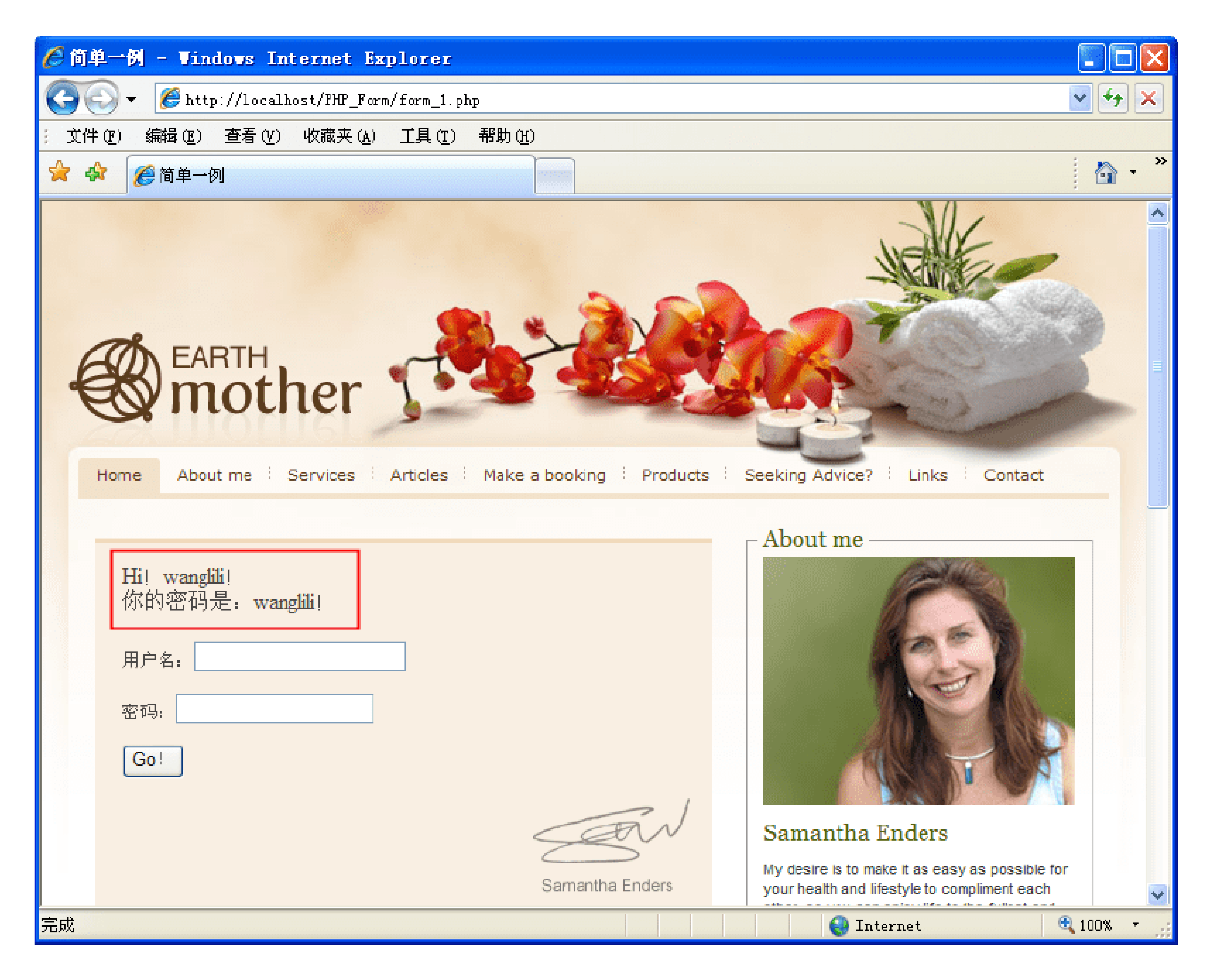


图 7-1 PHP 获取表单元素

7.1.4 实例分析


源码解析：

在上例中，条件逻辑指示 echo 语句只在用户提交(POST)了表单之后才发生，然后使用$_POST['username']可以获取表单元素 “username” 的值。

将表单提交回表单所在脚本时，可以使用 PHP 超级全局变量$_SERVER['PHP−−SELF']，执行脚本的名字会自动赋给此变量，因此，如果使用这个超级全局变量而不是实际的文件名，以后文件名有改变时，就不会额外地修改代码，从而能节省时间。例如，前面实例中的<form>标记可以改为<form action="<?php echo $_SERVER['PHP_SELF'];?> " method="post">，仍然会得到相同的结果。

7.2 使用带参数的函数获取邮箱地址

向函数传递表单数据的过程与传递其他变量的过程是一样的，只是将提交的表单数据作为函数参数传递即可。本节将为大家讲解如何使用 PHP 向函数传递表单数据。


视频教学：光盘\videos\07\使用带参数的函数获取邮箱地址.avi 长度：8 分钟

7.2.1 基础知识——向函数传递表单数据

如前面所述，向函数传递表单数据的过程与传递其他变量的过程是一样的，因此首先需要定义一个含有参数的函数，语法如下：

```
function validate_email($email)
{
    /*对参数$email 做一些验证处理，代码省略*/
}
```

接着需要向函数 validate_email()函数传递$email 参数，即调用 validate_email()函数，并传递一个参数给该函数。向函数传递表单数据的代码如下：

```
validate_email($_POST['email'])
```

上面代码是将表单元素为“email”的值传递给了 validate_email()函数，当然前提是 Action 采用的是 POST 提交方式。

7.2.2 实例描述

前面一节我们讲述了如何使用 PHP 获取“门”外的人，下面用一个实例来讲述一下如何对“门外的人”做检测，判断该“人”是否是善者。

很多时候我们在注册用户的时候，需要先输入邮箱地址，那么，是否随便输入一个邮箱地址就可以了呢？那岂不是“无米之炊”，因此我们不仅需要获取表单元素的邮箱地址内容，还应当对其进行一些验证，例如邮箱地址中是否有“@”符号和“.”符号之类的验证。那么，下面我们就来做一下吧！

7.2.3 实例应用

【例 7-2】 使用带参数的函数获取邮箱地址

在 form_2.php 页面中定义表单，代码如下所示：

```
<form action="form_2.php" method="post">
        <p>
            用户名：<input type="text" name="username" maxlength="20" />
        </p>
        <p>
            邮箱：<input type="text" name="email" maxlength="50" />
        </p>
        <input type="submit" name="submit" value="Go! " />
</form>
```

接着在 form_2.php 页面中使用 PHP 获取表单元素值并对其进行验证，代码如下：

```
<?php
function validate_email($email){
```

```
    $regexp="^([_a-z0-9]+)(\.[_a-z0-9]+)*@([a-z0-9]+)(\.[a-z0-9-]+)*(\.[a-z
]{2,6})$";
    //验证邮箱地址
    if(eregi($regexp,$email))
        return 1;
    else
        return 0;
}
if(isset($_POST['submit'])){
    echo "Hi!".$_POST['username']."<br/>";
    if(validate_email($_POST['email']))
    {
        echo "您的邮箱地址".$_POST['email']."是有效的";
    }
    else
        echo "您的邮箱地址<font color='red'>".$_POST['email']."</font>是无效的";
}
?>
```

7.2.4　运行结果

运行 form_2.php 页面，在表单中输入用户名和错误的邮箱地址，单击 Go！按钮，获取表单元素值，并对邮箱地址进行判断，提示邮箱地址无效信息，如图 7-2 所示。



图 7-2　验证邮箱地址无效

如果用户输入的地址有效，出现如图 7-3 所示的页面。



图 7-3　验证邮箱地址有效

7.2.5　实例分析


源码解析：

在上例中，我们使用了一个函数来确定电子邮件地址的有效性，然后使用 PHP 中的$_POST 变量获取表单中的邮箱地址内容，并传递给了上面我们定义好的函数中，从而对邮箱地址做了一些处理。

7.3　获取多个选项表单元素的值

多值表单组件(如复选框和多选框)大大提高了基于 Web 的数据收集能力，因为这些组件允许用户为一个指定的表单项同时选择多个值，那么使用 PHP 如何获取多值表单组件的值呢？这节我们为大家介绍。


视频教学：光盘\videos\07\处理多值表单组件.avi　　　　长度：6 分钟

7.3.1　基础知识——处理多值表单组件

多值表单组件在 Web 开发中是比较常用的，它能大大提高基于 Web 的数据收集能力，例如，一个表单，它能用来判断用户对计算机的哪些方面感兴趣。我们可以在表单域中编辑复选框，代码如下所示：

```
您对计算机哪方面感兴趣?
<input type="checkbox" name="languages[]" value="C#" /> C#
<input type="checkbox" name="languages[]" value="JavaScript" />JavaScript
<input type="checkbox" name="languages[]" value="Perl" />Perl
<input type="checkbox" name="languages[]" value="PHP" />PHP
```

因为这些组件是多值的，所以表单处理函数必须能够识别一个表单变量中可能有多个值。在前面的实例中，注意它们都使用了名为 languages 来引用多个语言选项。PHP 是怎么处理这种情况的呢？PHP 把它们考虑为数组。为了让 PHP 识别赋给一个表单变量的多个值(即考虑为数组)，需要对表单项名稍做一点修改，向其增加一对中括号。因此，表单项名将不是 languages，而是 languages[]。重命名后，PHP 将像处理所有其他数组一样对待所提交的变量，如下面的代码所示：

```
    foreach($_POST['languages'] AS $language) echo "$language";
```

这样，当用户提交表单后，将获取多值表单组件选中的值。

7.3.2　实例描述

大家都知道，当一年快要结束时，电视上会播放今年获得最喜爱的男歌手奖是 XXX，获得最佳女演员奖项的是 XXX……那么，我们试想一下，这些获奖的选手是怎么产生的呢？其实，就是通过网络等媒体产生的，人们可以选择自己心目中最喜爱的男歌手，当然可以选择多个，就像是一个人喜欢周杰伦，同时也很喜欢林俊杰一样，那么，下面就让我们来模拟一下投票功能吧！

7.3.3　实例应用

【例 7-3】 获取多个选项表单元素的值

编辑 form_3.php 页面，在页面中定义一个表单域，并在其中编辑 4 个复选框，并命名为“actress[]”，代码如下所示：

```
<form action="form_3.php" method="post">
        <p>
            您最喜爱的女演员是:
        </p>
        <p>
            <input type="checkbox" name="actress[]" value="刘亦菲" />刘亦菲
        </p>
        <p>
            <input type="checkbox" name="actress[]" value="林心如" />林心如
        </p>
        <p>
            <input type="checkbox" name="actress[]" value="范冰冰" />范冰冰
        </p>
        <p>
```

```
            <input type="checkbox" name="actress[]" value="赵薇" />赵薇
        </p>
        <input type="submit" name="submit" value="Go! " />
</form>
```

当表单提交时，获取用户选中的复选框的值，并在页面中输出，代码如下：

```
<?php
        if(isset($_POST['submit'])){
                    echo "您最喜爱的女演员是：<br/>";
                    foreach($_POST['actress'] AS $actress) echo "$actress<br/>";
        }
?>
```

7.3.4　运行结果

运行 form_3.php 页面，选择喜爱的女演员，单击 Go！按钮，提交至本页面，获取选中的值，并在 form_3.php 页面中输出，如图 7-4 所示。



图 7-4　处理多值表单组件

7.3.5　实例分析



源码解析：

在上例中，我们将复选框的 name 属性赋值为“actress[]”，而并非是“actress”，这样，PHP 把表单中的多个复选框的值当做一个数组来对待。接着循环数组，将数组中的元素赋值给 actress 变量，并使用$actress 输出数组中的值，从而获取了选中的复选框的值。

7.4　使用 PHP 动态生成表单元素

许多基于 Web 的表单需要做更多工作，而不只是简单地组装几个域。诸如复选框、单选按钮和下拉列表框等表单项都非常有用，它们可以向表单增添很多功能。但是，通常是希望根据一些来自动态源(如数据库)的数据为这些表单项赋值。PHP 可以轻而易举地完成这项任务，本节将为大家介绍如何使用 PHP 生成表单元素。

 **视频教学：** 光盘\videos\07\使用 PHP 动态生成表单元素.avi　　长度：7 分钟

7.4.1　基础知识——用 PHP 生成表单元素

在很多情况下，需要使用 PHP 根据不同的请求从数据库读取数据并生成表单元素，那么在 PHP 开发中常用到的表单元素有哪些呢？如何使用 PHP 自动生成表单元素？这些将在本节介绍。

1. 表单元素

一个表单是由多个表单元素构成的，表单所使用的标签元素有十几个，PHP 开发中常用及较重要的表单元素如表 7-1 所示。

表 7-1　表单元素

表单元素	说　明
input type="checkbox"	复选框，允许用户选择多个选择项
input type="file"	文件浏览框，当文件上传时，可用来打开一个模式窗口以选择文件
input type="hidden"	隐藏标签，用于在表单中以隐含方式提交变量值
input type="password"	密码文本框，用户在该文本框输入字符时将被替换显示为*号
input type="radio"	单选项，用于设置一组选择项，用户只能选择一个
input type="reset"	清除与重置表单内容，用户清除表单中所有文本框的内容，而且使选择菜单项恢复到初始值
input type="submit"	表单提交按钮
input type="text"	单行文本框
select	下拉列表框，可单选和多选。默认为单选，如果增加多项选择功能，增加<select name="select" size="自定义列数"multiple="multiple">即可
option	列表下拉菜单，和 select 配合使用，显示供选择的值
textarea	多行文本框，在使用文本框时需要关闭标签之间的文本内容，形成如下格式：<Textarea>文字</Textarea>

其中，hidden 标签被称为隐藏或隐含的标签，它不会在用户浏览的界面中出现，当用户填写资料表单和跨页之间传值时，可以使用该标签传递一些隐含的值。

password 密码文本框用于隐藏密码，用户输入的文本将以*显示在文本框中，但是密码并没有加密，只是被*替换显示。

下面介绍一下表单的属性，它们用于表单中约束表单元素的行为或显示，其含义与说明如表 7-2 所示。

表 7-2　表单元素属性

属性名称	说　明
name	文本框的名称，PHP 根据该名称，在超级全局数组中建立以 name 为名称的键名
size	文本框的宽度，在 select 下拉菜单中，表示可以看到的选项行数
value	文本框中的默认值，注意，该值不能应用到 type=password 密码文本框，以及 type=file 文件文本框中
multiple	此属性用于下拉列表菜单 select 中，指定该选项用户可以使用 Ctrl 和 Shift 键进行多选
rows	多行文本框显示时可以容纳的字符列数宽度
cols	多行文本框显示时可以容纳的字符行数高度

除了以上一些必要的属性元素外，还有一些标准属性，如 class、style、id 等，可以参阅 HTML 相关资料做进一步了解，这里不再赘述。

2. 使用 PHP 动态生成表单元素

在一些动态脚本中，需要使用 PHP 根据不同的请求从数据库生成表单元素，而页面中的表单元素最主要的有 3 个属性，分别为 name、value 及默认选项，因此使用 PHP 自动生成表单元素时，需要定义的函数中一般接受以下这 3 个参数。

```
function radio($name, $options, $default="")
```

其中：

- $name：赋给表单元素的名称，即 name 属性，确定如何引用提交的变量。
- $options：用于赋给表单元素的值，即 value 属性。如果是下拉列表框，其包含用户创建选择菜单项的键值对。
- $default=""：默认为空，即表单元素的默认选项。

使用 PHP 自动生成表单元素和使用其他脚本语言生成表单元素的原理是一样的，都需要使用 PHP 语言生成 HTML 页面脚本语言，例如自动生成复选框组，代码如下：

```
//循环遍历表单元素选项
foreach($options as $value => $label)
{
    //使用 PHP 语言生成页面脚本
    $html .= "<input type=checkbox ";
    //向复选框的 name 属性和 value 属性赋值
    $html .= "name=\"{$name}[]\" value=\"$value\">";
    $html .= $label . "<br>";
}
```

下面就来展示几种生成表单按钮或选项的方法。

1)　动态生成一组单选按钮

使用 PHP 语言在页面中生成一组单选按钮的代码如下：

```
<?php
        //定义单选按钮项
       $options = array(
            "010" => "北京",
            "020" => "上海",
            "024" => "沈阳",
            "0411"=> "大连"
        );
        //定义默认选项为“沈阳”
        $default = "024";
        //向html赋值，调用generate_radio_group($name, $options, $default="")函数
        $html = generate_radio_group("city_id", $options, $default);
        echo $html;
        //generate_radio_group($name, $options, $default="")函数
        function generate_radio_group($name, $options, $default="")
        {
            //向表单元素 radio 的 name 属性赋值为“city_id”，一组单选按钮的 name 属
              性值是相同的
            $name = htmlentities($name);
            //循环遍历单选按钮项
            foreach($options as $value => $label)
            {
                //向表单元素 radio 的 value 属性赋值为上面定义好的 options 数组元素
                $value = htmlentities($value);
                //使用 PHP 语言生成 HTML 页面表单元素代码
                $html .= "<INPUT TYPE=\"RADIO\" ";
                if ($value == $default)
                {
                    $html .= "CHECKED ";
                }
                $html .= "NAME=\"$name\" VALUE=\"$value\">";
                $html .= $label . "<br>";
            }
            //把生成的表单元素返回
            return($html);
        }
?>
```

该脚本将生成一列单选按钮组，名称为“city_id”，默认选项为“沈阳”。在浏览器中查看生成的单选按钮组，如图 7-5 所示。



图 7-5　使用 PHP 自动生成单选按钮组

2)　动态生成复选框组

使用 PHP 语言在页面中自动生成一组复选框选项的代码如下：

```
<?php
        //定义生成复选框组的函数
    function generate_checkboxes($name,$options, $default=array())
        {
            if (!is_array($default))
            {
                $default = array();
            }
            //循环遍历复选框选项
            foreach($options as $value => $label)
            {
                //使用 PHP 语言生成页面脚本
                $html .= "<input type=checkbox ";
                if (in_array($value, $default))
                {
                    $html .= "checked ";
                }
                //向复选框的 name 属性和 value 属性赋值
                $html .= "name=\"{$name}[]\" value=\"$value\">";
                $html .= $label . "<br>";
            }
            return($html);
        }
        //定义复选框选项值
        $interests = array("音乐"  => "音乐","电影" => "电影","互联网"  => "
互联网","旅游"  => "旅游");
        //生成 html 页面脚本
        $html = generate_checkboxes("interests",$interests, "");
    ?>
```

```
        选择您的爱好：<form action="form_checkbox.php" method=post>
         <?php
             echo $html;
         ?>
         <input type=submit value="确定"></form>
```

在页面中生成了一组复选框，名称为“interests”，选项分别为音乐、电影、互联网和旅游。在浏览器中查看在页面中生成的复选框组，如图 7-6 所示。



图 7-6　使用 PHP 自动生成复选框组

3)　生成多选下拉列表菜单

使用 PHP 在页面中生成多选下拉列表菜单项的代码如下：

```
<?php
            //定义下拉列表菜单选项，键值对存在
            $options = array(
                '1' => '请选择',
                'news' => '新闻',
                'events' => '事件',
                'publications' => '稿件'
            );
            //设置默认选项为“新闻”
             $default = "news";
             //生成多选下拉列表项表单元素
             $html =generate_muilti_option("select", $options, $default);
             echo $html;
             //定义生成多选下拉列表菜单元素的函数
             function generate_muilti_option ($name,$options, $default)
             {
                 //建立一个允许多选的列表单
                 echo '<select name="'.$name.'[]" id="'.$name.'[]" multiple=
"multiple">';
                 foreach( $options as $value => $option )
                 {
```

```
                    //向多选下拉列表菜单元素赋值
                    echo '<option value="' . htmlspecialchars( $value ) . '"' ;
                    if( $default == $value )
                    {
                        echo ' selected';
                    }
                    echo '>' . htmlspecialchars( $option ) . '</option>';
                }
                echo '</select>';
            }
?>
```

该脚本将生成一个可多选的下拉列表项表单元素，元素名称为“select[]”，选项有 4 个，分别为请选择、新闻、事件和稿件，默认选项为新闻。在浏览器中浏览生成的多选下拉列表菜单效果，如图 7-7 所示。



图 7-7　使用 PHP 生成多选下拉列表菜单

一般动态生成的表单元素，多为从数据库取得数据或数据数组，转换成动态的 HTML 表单元素，也可以手动创建。

7.4.2　实例描述

商家最关心的是产品是否是客户喜欢的，因此就产生了社会调查系统。我一个做家电买卖的朋友，非常注重客户的需求，于是让我给他做一个调查页面，调查每个人喜欢的家电产品及需求。下面就来做一个调查页面。

7.4.3　实例应用

【例 7-4】 使用 PHP 动态生成表单元素

创建 form_4.php 页面，在页面中使 PHP 与 MySQL 数据库连接，并读取数据库表中的数据作为表单元素的属性值，使 PHP 自动生成表单元素。完成代码如下：

```
<form action="form_4_submit.php" method="post">
        <?php
            //创建连接 mysql 数据库信息
            $conn=mysql_connect("localhost","root","000") or die("Can't connect
to database");
            //设置要连接的数据库为“product”数据库
            mysql_select_db("product") or die("Can't select database");
            //定义 sql 语句，查询数据库表信息
            $query="select id,name from table_product order by id";
            //获取结果
            $result=mysql_query($query);
            //向变量 name 赋值
            $name =htmlentities( "city_id");
            while($row=mysql_fetch_array($result)){

                    //向表单元素 radio 的 value 属性赋值从数据库表中查询的“name”字段
                      的值
                    $value = htmlentities($row["name"]);
                    //使用 PHP 语言生成 HTML 页面表单元素代码
                    $html .= "<INPUT TYPE=\"RADIO\" ";
                    $html .= "NAME=\"$name\" VALUE=\"$value\">";
                    $html .=$row["name"];
                    $html .= $label . "<br>";

            }
            echo $html;
        ?>
        <center><input type="submit" value="提交" /></center>
       </form>
```

接着编辑 form_4.php 页面提交页面 form_4_submit.php，在页面中获取选中的表单元素的值。代码如下：

```
<?php
                print_r( "Your favorite products:".$_POST['city_id']);
?>
```

7.4.4　运行结果

运行 form_4.php 页面，在页面中输出一组单选按钮，如图 7-8 所示。

图 7-8　使用 PHP 动态生成单选按钮组

单击图 7-8 中的【提交】按钮，提交至 form_4_submit.php 页面，在页面中输出选中用户选择的最喜爱的产品信息内容，如图 7-9 所示。

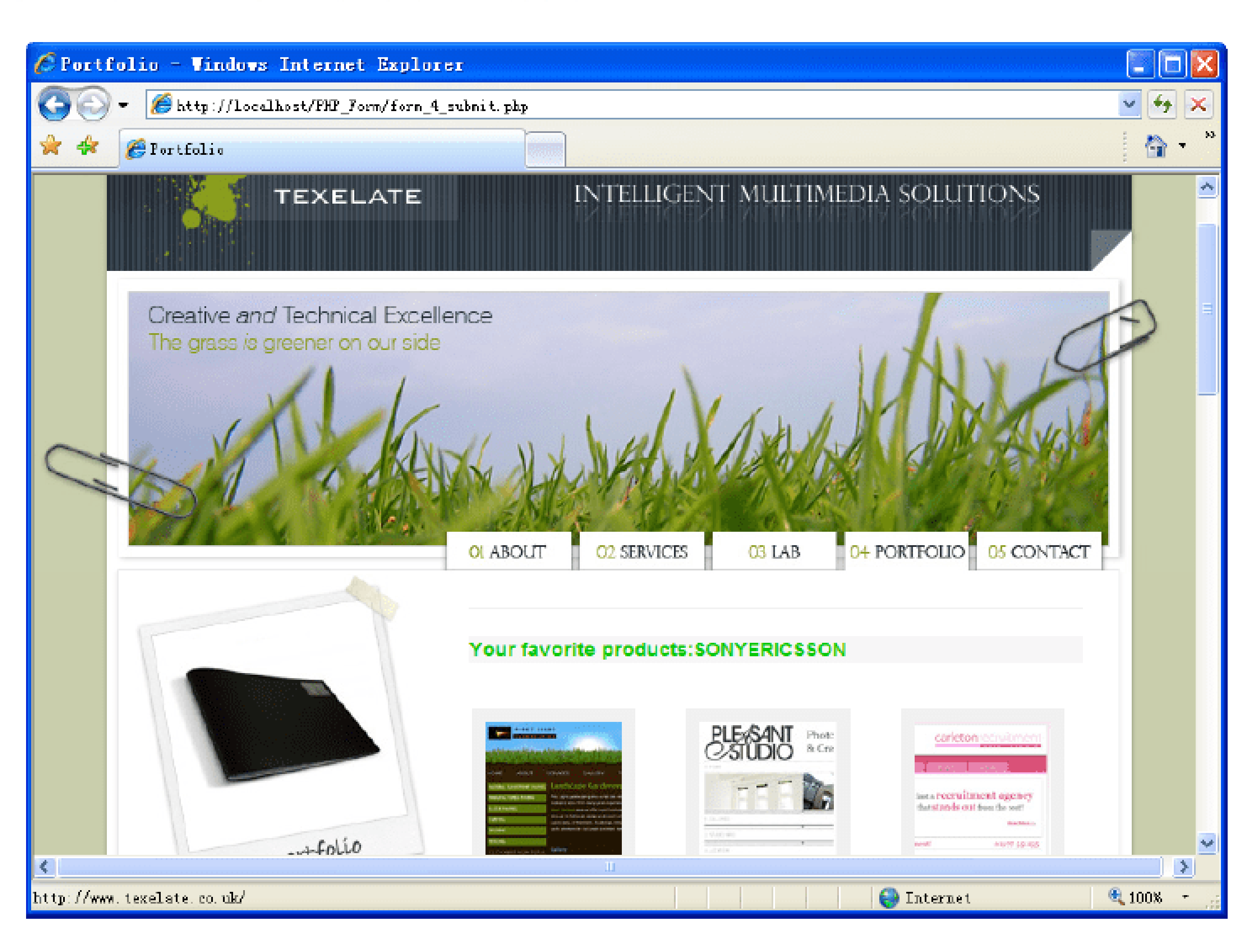


图 7-9　获取表单元素的值

7.4.5　实例分析


源码解析：

在上例中，我们使用 PHP 语言连接 MySQL 数据库，动态地读取数据库表 table_product 中的数据，并作为表单元素(单选按钮组)的值，这是在开发中经常用到的一种技术。使用

"$row=mysql_fetch_array($result)" 获取到从数据表中查询出来的信息，它是一个键值对的集合对象，从而使用$row['字段名称']可获取字段值。

7.5　制作能自动选择预定值的家电调查页面

高质量的 GUI 设计在很大程度上取决于一致性。也就是说，整个网站中使用协调的可视界面，尤其是与用户直接交互的组件(例如表单)，这是个不错的想法。为了有利于建立一致的界面，最好尽可能地重用基于表单的代码，另外，要对数据的插入和修改采用相同的模板。当然，你可能认为这样的策略会很快导致逻辑和表现的混乱。但是，只要事先有所考虑，不仅能鼓励代码重用，还能保证一些好的编程实践，这实际上很简单。本节将介绍这样一种方法。

 **视频教学**：光盘\videos\07\自动选择表单数据.avi　　长度：6 分钟

7.5.1　基础知识——自动选择表单数据

上一节展示了如何创建一个通用函数来创建动态生成的单选按钮组。为了展示本节介绍的概念，下面来看一个例子。

假设一个网站提供了一个注册表单，提示用户的首选语言以及其他内容。该语言将作为将来电子邮件的默认语言。但是，语言的选择取决于支持人员的语言能力，而这是人力资源部所管理的记录。因此，并非碰运气一样地提供一个过期的可用语言列表，而是要把用于这个表单项的下拉列表直接链接到人力资源部所用的语言表。此外，因为下拉列表的每个元素由 3 个部分组成(标识列表本身的名称以及每个列表项的值和名)，所以可创建一个函数来抽象这个任务，使之既能生成动态列表，又能自动选择预定的值，要想实现这样的功能，必须提供一个参数，比如$key。那么可以定义一个名为 create_dropdown()的函数，它接受 5 个输入参数。

- $identifier：赋给下拉列表菜单的名称，确定如何引用提交的变量。
- $pairs：一个关联数组，其包含用于创建选择菜单项的键-值对。
- $firstentry：作为下拉列表菜单的可视化提示，将放在第一个位置。
- $multiple：该下拉列表菜单允许多选吗？如果允许，则传入 multiple；否则不传入任何值(此参数可选)。
- $key 这个可选参数保存自动选择的元素。如果未赋值，则不会自动选择任何值。

此函数在构建下拉列表菜单时，将每个元素与$key 比较，确定某个元素是否应当自动选择。为了使代码更简练，比较时使用了三元运算符。函数代码如下：

```
<?php
    function
create_dropdown($identifier,$pairs,$firstentry,$mutiple="",$key)
    {
        $dropdown="<select name=\"$identifier\"multiple=\"$multiple\">";
        $dropdown.="<option name=\"\">$firstentry</option>";
        foreach($pairs AS $value=>$name)
        {
```

```
            $dropdown.=($value=$key)?
                        "<option name=\"$value\" selected=\"selected\">$name
</option>":
                        "<option name=\"$value\">$name</option>";
        }
        echo "</select>";
        return $dropdown;
    }
?>
```

假设从数据库中读取的信息有 4 条数据，分别是 Dutch、English、Italian 和 Spanish，这 4 条信息作为了下拉列表菜单的选项。如果希望下拉列表菜单自动选择语言为 Italian，只要将其相应的标识符(例如“2”)传入即可，代码如下：

```
echo create_dropdown("language",$pairs,"Choose One","",2);
```

这将得到如下的输出：

```
Choose  your preferred language:<br>
<select name="language">
    <option name="">Choose One</option>
   <option name="1">English</option>
   <option name="2" selected="selected">Italish</option>
   <option name="3">Spanish</option>
   <option name="4">Dutch</option>
</select>
```



这里的 Italian 选项已被选中。细心的同学应该注意到，在前面讲述使用 PHP 自动生成表单元素的一节中，有一段使用 PHP 语言生成单选按钮组的代码，在那段代码中，已经实现了自动选择表单数据的功能，和这里的思路是一样的。

7.5.2 实例描述

上述为朋友做的家电调查页面，不是很完善，如果用户没有选择任何家电产品也可以提交页面，这样造成了很多无效数据，给服务器和数据库造成了很大的压力，因此，就改善了一下，下面就来看看改善后的页面程序。

7.5.3 实例应用

【例 7-5】 制作能自动选择预定值的家电调查页面

编辑 form_5.php 页面，在页面中动态生成单选按钮组，并实现能自动选择预定值的功能，代码如下：

```
<form action="form_4_submit.php" method="post">
       <?php
           //创建连接 mysql 数据库信息
           $conn=mysql_connect("localhost","root","000") or die("Can't connect
```

```
to database");
        //设置要连接的数据库为“product”数据库
        mysql_select_db("product") or die("Can't select database");
        //定义 sql 语句，查询数据库表信息
        $query="select id,name from table_product order by id";
        //获取结果
        $result=mysql_query($query);
        //generate_radio_group($name, $options, $default="")函数
        function generate_radio_group($name, $options,$firstentry, $key)
        {
            //向表单元素 radio 的 name 属性赋值为“city_id”，一组单选按钮的 name 属
              性值是相同的
            $name = htmlentities($name);
            //循环遍历单选按钮项
            foreach($options as $value => $label)
            {
                //向表单元素 radio 的 value 属性赋值为上面定义好的 options 数组元素
                $value = htmlentities($value);
                //使用 PHP 语言生成 HTML 页面表单元素代码
                $html .= "<INPUT TYPE=\"RADIO\" ";
                $html.=($value==$key)?
                        " checked=\"checked\" ":"";
                $html .= "NAME=\"$name\" VALUE=\"$value\">";
                $html .= $label . "<br>";
            }
            //将生成的表单元素返回
            return($html);
        }

        //向变量 name 赋值
        $name =htmlentities( "city_id");
        while($row=mysql_fetch_array($result)){

                //向表单元素 radio 的 value 属性赋值从数据库表中查询的“name”字段的值
                $value = $row["name"];
                $id=$row["id"];
                $pairs["$id"]=$value;

        }
        echo generate_radio_group($name, $pairs,"Choose One", 2);
?>
```

7.5.4　运行结果

运行 form_5.php 页面，在页面中动态生成一组单选按钮，并实现了自动选择预定值的功能，如图 7-10 所示。

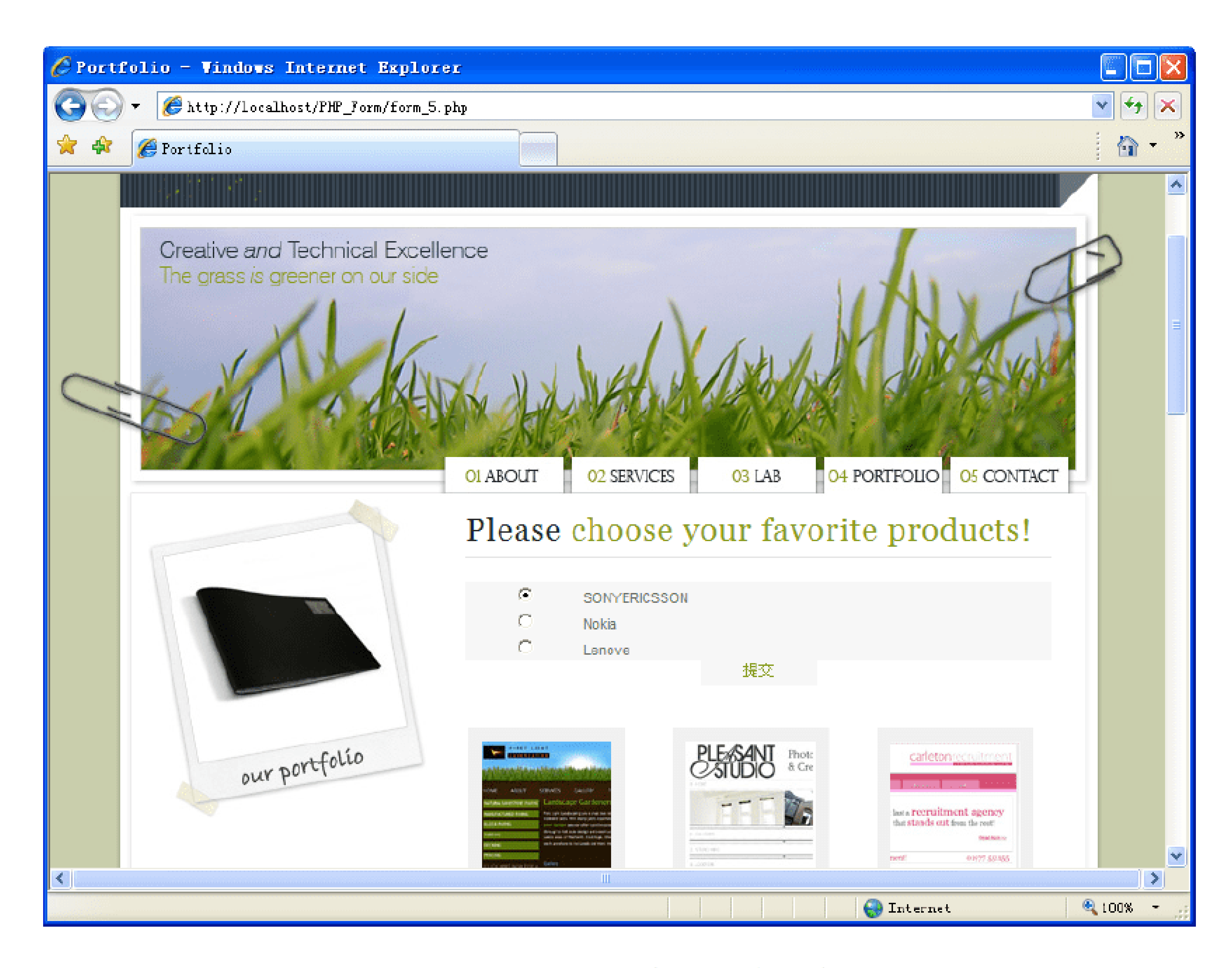


图 7-10 自动选择预定值的单选按钮组

7.5.5 实例分析


源码解析：

在上例中，动态的从数据库表中读取了数据，以 while 循环遍历了数据集，其中，while 循环块中的$pairs 变量是一个以键-值对存在的集合。上例中的 generate_radio_group($name, $options,$firstentry, $key)函数实现了自动生成表单元素的功能并自动选择预定值的功能，函数中的$key 参数非常关键，用来标识选中状态，最后向 generate_radio_group($name, $options,$firstentry, $key)函数传递参数，并把获取到的$pairs 变量传递到该函数中，向生成的表单元素的 value 属性赋值。

7.6 使用 PHP 与 JavaScript 交互实现 Cookie 功能

使用 PHP 作为主要的脚本语言并不意味着就应当依赖它做的所有事情。事实上，如果结合使用 PHP 和客户端的语言，如 JavaScript，通常会极大地拓展应用程序的灵活性。但是，对此有一个常见的问题：这两种语言如何对话，因为 JavaScript 在客户端执行，而 PHP 在服务器端执行。这节将讲述 PHP 与 JavaScript 之间的交互。


视频教学：光盘\videos\07\PHP、Web 表单和 JavaScrip.avi　　长度：5 分钟

7.6.1　基础知识——PHP、Web 表单和 JavaScript

许多网站都提供了这样一个功能，可以通过电子邮件向朋友发送文章或新闻故事。有时是通过“弹出”窗口实现的，而“弹出”窗口会提示用户输入接受方的地址及其他信息。提交表单后，文章将邮寄到接收方，接下来用户关闭窗口。通常，弹出动作是使用 JavaScript 完成的，而邮件提交则使用 PHP 完成。但是，由于 JavaScript 要打开新窗口，所以必须能够传递一些重要信息，如唯一的文章标识符(它能唯一的标识文章)。

如下脚本演示了如何完成这个任务。由此可以看出，向 JavaScript 函数传递 PHP 变量是多么简单。在文档开头，定义了一个名为 mail()的 JavaScript 函数。这个函数打开一个指向 PHP 脚本的固定大小窗口，再由这个 PHP 脚本给出提示，并处理提交邮件。

```
<html>
<head>
<title>Breaking News</title>
</head>
<script type="text/javascript">
    function mail(id){

    window.open("mail.php?id="+id,"info","width=250,height=250,scrollbars=0
,resizable=0");
    }
</script>
<body  bgcolor="#FFCCCC"  text="#000000"  link="#0000ff"  vlink="#800080"
alink="#ff0000">
<a href="#" onclick="mail(<?php echo $id;?>);">
    //寄这篇文章给一位朋友
</a>
//文章的内容在这里……
</body>
</html>
```

运行页面，将打开一个类似于图 7-11 所示的页面。单击链接后，将会打开“mail.php”页面。

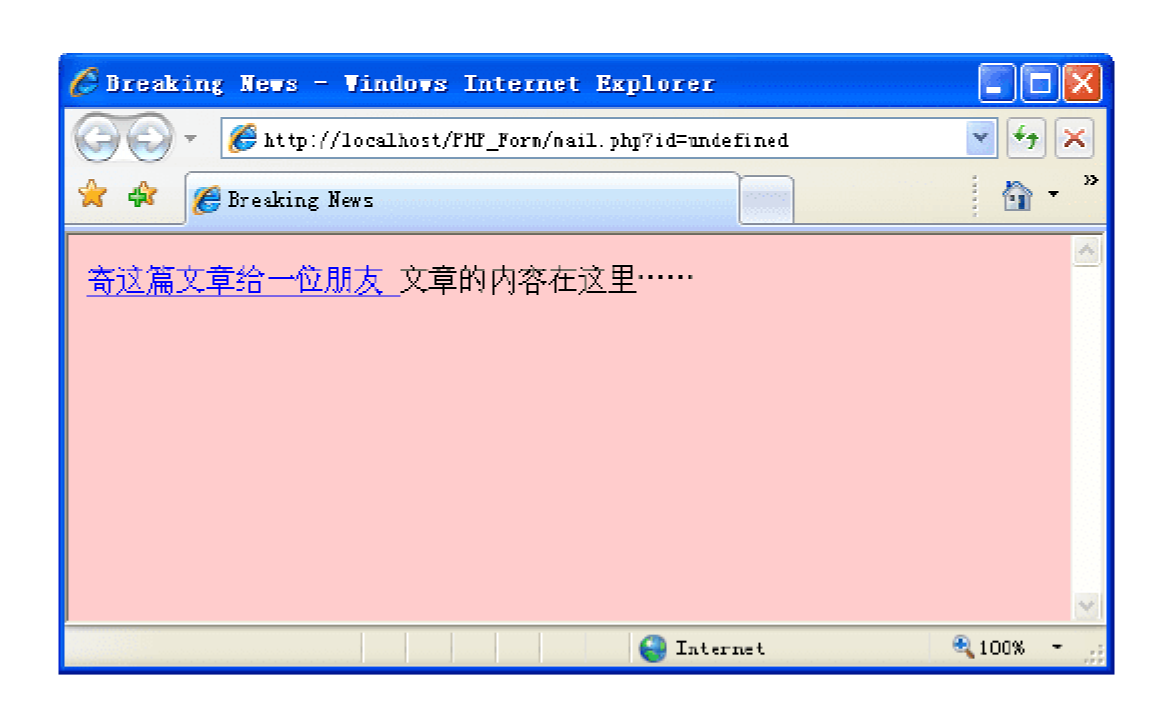


图 7-11　PHP 和 JavaScript 交互显示界面

通过上面的案例，应该明白 PHP 与 JavaScript 之间的交互原理了。和其他语言一样，都是需要使用语言脚本输出一个变量传递给 JavaScript 函数作为参数即可。

7.6.2 实例描述

Cookie 是由服务器端生成，发送给 User-Agent(一般是浏览器)，浏览器会将 Cookie 的 key/value 保存到某个目录下的文本文件内，下次请求同一网站时就发送该 Cookie 给服务器(前提是浏览器设置为启用 Cookie)。Cookie 名称和值可以由服务器端开发自己定义，对于 JSP 而言可以直接写入 jsessionid，这样服务器可以知道该用户是否为合法用户，以及是否需要重新登录等。下面使用 PHP 与 JavaScript 交互做一个类似于 Cookie 的功能。

7.6.3 实例应用

【例 7-6】 使用 PHP 与 JavaScript 交互实现 Cookie 功能

(1) 编辑 action.html 页面，在页面中使用框架将其他页面整合。完整代码如下：

```
<html>
<head>
<title>获取 Cookie 信息</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<frameset framespacing="0" border="false" frameborder="0" cols="0,*">
  <frame name="leftFrame" scrolling="no" noresize src="action.php">
  <frame name="rightFrame" scrolling="auto" src="main.html">
</frameset><noframes>
  <body bgcolor="#FFFFFF">
  <p>本页使用页面框架，但是您的浏览器不支持。</p>
  </body>
</noframes>
</html>
```

(2) 编辑 action.php 页面，将随机生成的整数放入 Cookie 中。完整代码如下：

```
<html>
<head>
<title>获取 Cookie 信息</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body bgcolor="#FFFFFF">
<?php
    srand((double)microtime()*1000000);
    $result=rand(0,100);
?>
<script language="javascript">
  parent.rightFrame.myform.current_cookie.value="<?php echo $result ?>";
</script>
```

```
</body>
</html>
```

(3)　编辑 main.html 页面，通过读取 Cookie 来实现交互。在 main.html 页面中也可以通过重新读取 action.php 页面来实现控制后台程序。完整代码如下：

```
<html>
<head>
<title>获取 Cookie 信息</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body bgcolor="#FFFFFF">
<form name="myform" >
当前参数为<input type="text" name="current_cookie" size="80" maxlength="1000">
</form>
<br>
<a href="action.php" target="leftFrame">重新读取 Cookie</a>
</body>
</html>
```

7.6.4　运行结果

运行 action.html 页面，效果如图 7-12 所示。



图 7-12　获取 Cookie 信息

7.6.5　实例分析



源码解析：

上例中总共包含了 3 个页面，分别为 action.html 页面、action.php 页面和 main.html 页面，前台 main.html 页面和后台 action.php 页面通过页面框架 action.html 页面组织起来，将 action.php

页面的宽度设为 0，这样并不影响显示。action.php 将信息放入了 Cookie 中，main.html 页面通过读取 Cookie 来实现交互，原理为通过 parent.rightFrame.myform.current_cookie.value 直接将 PHP 生成的随机数传递到 main.html 页面的文本框中显示出来。

7.7 使用基于文件的身份认证实现登录验证

用户在设计和维护站点的时候，经常需要限制对某些重要文件或信息的访问。通常，我们可以采用内置于 Web 服务器的基于 HTTP 协议的用户身份验证机制。当访问者浏览受保护页面时，客户端浏览器会弹出对话窗口要求用户输入用户名和密码，对用户的身份进行验证，以决定用户是否有权访问页面。PHP 提供了两种身份认证的方式：基本的 HTTP 身份认证和 PHP 身份认证。本节将详细介绍这两种身份验证。

视频教学：光盘\videos\07\HTTP 身份验证.avi　　长度：3 分钟
光盘\videos\07\PHP 身份验证.avi　　长度：10 分钟

7.7.1 基础知识——基本的 HTTP 身份验证

HTTP 身份认证功能是指客户端浏览器弹出一个会话框，要求用户输入用户名和密码。这种认证方式在许多会员制网站和对用户浏览模式进行调查的站点中应用十分普遍。用户还可以在网站上增设一些管理网页，也要使用身份认证来保证只有管理员才可以进入，而其他用户访问不到这些网页。

HTTP 身份认证功能可以通过 PHP 的 Header()函数向客户端浏览器发送一个“Authentication Required”的消息，浏览器便会弹出一个会话框，要求客户输入用户名及密码，通过验证后服务器端将会再次调用后续的 PHP 程序，并将用户输入的用户名和密码分别赋给 PHP 的内置变量$PHP_AUTH_USER 和$PHP_AUTH_PW，将身份认证类型赋给变量$PHP_AUTH_TYPE。下面来做一个实例，代码如下：

```
<?php
    if (!isset($_SERVER['PHP_AUTH_USER'])){ //如果$PHP_AUTH_USER 变量没有被赋值
        //在会话框的"领域"项后面显示的文字
      Header("WWW-Authenticate: Basic realm=\"汇智科技解答\"");
      Header("HTTP/1.0 401 Unauthorized");
        //当用户按下了"取消"按钮便显示如下信息，并退出
        echo $_SERVER['PHP_AUTH_USER']."你好<P>";
      echo "你取消了登录！";
      exit;
    } else {
        if(!($_SERVER['PHP_AUTH_USER']=="admin"&&
$_SERVER['PHP_AUTH_PW']=="admin"))
        {
            //如果错误的用户名或密码，强制再验证
            Header("WWW-Authenticate: Basic realm='请输入正确的用户名和密码'");
```

```
            Header("HTTP/1.0 401 Unauthorized");
            echo "你输入的用户名或密码错误，请重新输入！";
            exit();
        }
        else
        {
            //向用户显示欢迎信息
             echo $_SERVER['PHP_AUTH_USER']."你好<P>";
             echo "请记住你的密码：".$_SERVER['PHP_AUTH_PW'];
        }
    }
?>
```

由于 IE 浏览器对 HTTP 首部数据的顺序要求很严格，所以在使用 Header()函数时，必须在发送“HTTP/1.0 401”的首部数据之前发送“WWW-Authenticate”首部数据。当用户输入用户名和密码后，Netscape 或 IE 会把用户身份认证信息自动缓存在浏览器所在的客户端。当再次遇到 HTTP 401 首部数据时，用户不需要再次输入密码，浏览器便自动将上次输入的密码发送到服务器端，该用户的身份认证信息直到关闭浏览器后才被清除。

从上面的代码可以看出，当应用程序取得了$PHP_AUTH_USER 和$PHP_AUTH_PW 两个变量之后，可以通过编写程序来判断用户名和密码正确与否，并可完成基于用户权限的操作。例如使用用户名和密码去查找数据库中的合法用户记录，或针对不同的用户显示不同的信息等。

7.7.2　基础知识——PHP 身份认证

将用户认证直接集成到 Web 应用程序逻辑中是很方便的，也很灵活。本节将主要介绍 PHP 的内置认证功能，并讲解一些可以立即集成到应用程序中的认证方法。

1. 基于 PHP 的基本认证

PHP 使用两个预定义的变量来认证用户，$_SERVER['PHP_AUTH_USER']和$_SERVER ['PHP_AUTH_PW']这两个变量保存了认证需要的两个部分，分别是用户名和密码，但是在使用这两个变量时应注意以下方面。

- 两个变量都必须在每个受限页面的开始处验证，这一点通过包装各个受限页面就能轻松地完成。用户可以将认证代码放在单独的文件中，然后在使用时通过 require()函数将该文件包含进来即可。
- 这些变量在 CGI 版本的 PHP 中不能正常工作，在 Microsoft IIS 上也不起作用。

技术文档	PHP 认证和 IIS
如果使用 PHP 语言和 IIS 结合，需要使用 PHP 的 HTTP 认证功能时，在做本章的示例时就需要对示例做一些修改。使用 IIS 时，用户名和密码变量在 PHP 中仍然可用，但不是通过$_SERVER['PHP_AUTH_USER']和$_SERVER['PHP_AUTH_PW']。事实上，这些值必须由另一个服务器全局变量$_SERVER['HTTP_AUTHORIZATION']来解析。例如，在需要使用到用户名和密码时，就需要以下的解析：isset($_SERVER['HTTP_AUTHORIZATION']	

在 PHP 处理认证时，常用到两个标准函数，分别是 header()函数和 isset()函数。

1) header()函数

在 PHP 中，使用 header()函数向浏览器发送一个原始的 HTTP 标头，该函数的语法格式如下所示：

```
void header (string string [, bool replace [, int http_response_code]])
```

header()函数有 3 个参数。

- string：该参数是必需的，规定要发送的报头字符串。
- replace：该参数是可选的，指示该报头是否替换之前的报头，或添加第二个报头。默认是 true(替换)。如果指定该参数为 false，表示允许相同类型的多个报头。
- http_response_code：该参数是可选的，表示将 HTTP 响应代码强制为指定的值。



在使用该函数时，需要在任何实际的输出被发送之前调用 header()函数，并且从 php 4.4 之后，该函数防止一次发送多个报头，这是对头部注入攻击的保护措施。

在调用 header()函数时，有两种特殊的调用方式。

第一种是标头以字符串“HTTP/”(大小写不重要)开头的，可以用来确定要发送的 HTTP 状态码。例如，如果配置了 Apache 用 PHP 来处理找不到文件的错误处理请求(使用 ErrorDocument 指令)，需要确保脚本产生了正确的状态码。

```
<?php
    header("HTTP/1.0 404 Not Found")
?>
```



HTTP 状态码标头行总是第一个被发送到客户端，而并不管实际的 header()函数调用是否是第一个。除非 HTTP 标头已经发送出去，任何时候都可以通过用新的状态行调用 header()函数来覆盖原先的。

第二种特殊情况是以“Location:”标头。它不只是把这个标头发送回浏览器，它还将一个 REDIRECT(302)状态码返回给浏览器，除非之前已经发出了某个 3xx 状态码。

```
<?php
    header("Location: http://www.example.com/"); /* 重定向浏览器 */
    /* 确保重定向后，后续代码不会被执行 */
    exit;
?>
```

2) isset()函数

在 PHP 中，使用 isset()函数检测变量是否设置，该函数的语法格式如下所示：

```
bool isset (mixed var [, mixed var [, ...]])
```

如果 var 存在则返回 true，否则返回 false。如果已经使用 unset()函数释放了一个变量之后，它将不再是 isset()函数。若使用 isset()函数测试一个被设置成 NULL 的变量，将返回 false。同时要注意的是一个 NULL 字节("\0")并不等同于 PHP 的 NULL 常数。



isset()函数只能用于变量，因为传递任何其他参数都将造成解析错误。若想检测常量是否已设置，可使用 defined()函数。

2. 基于文件的身份认证

通常需要为每个用户提供唯一的登录对，这样可以记录用户特定的登录时间、活动和动作。就像存储 Unix 用户有关信息常用的做法一样，可以利用文本文件轻松实现，其中每行包含一个用户名和加密密码对，它们之间用冒号(:)分隔，user.txt 文件内容如下：

```
sa:605de8457e
maxianglin:58455a54edc
```



这里需要注意的是，user.txt 文件应当存储在服务器文档根目录之外。否则，攻击者有可能通过一些不正常手段来获取该文件并查看该文件的内容，从而造成信息的泄露。此外，用户可以选择不对密码加密，以明文形式存储。但这种方式极不安全，因为如果文件权限配置不当，那么能够访问服务器的用户就可以查看这些登录信息。

要解析 user.txt 文件并根据给定的登录对来认证用户，为此所需的 PHP 脚本只比根据硬编码认证对完成认证所用的脚本稍微复杂一点。区别在于此脚本必须将文本文件读取到数组中，然后循环处理数组，搜索匹配的数据来进行身份认证。在这个过程中需要使用以下函数。

- file (string filename [, int use_include_path [, resource context]])：该函数将整个文件读取到数组中，数组的每个单元都是文件中相应的一行，包括换行符在内。
- explode (string separator, string string [, int limit])：该函数将一个字符串分隔另一个字符串，每个元素都是 string 的一个子串，它们被字符串 separator 作为边界点分隔出来。如果设置了 limit 参数，则返回的数组包含最多 limit 个元素，而最后那个元素将包含 string 的剩余部分。
- md5 (string str [, bool raw_output])：该函数使用 RSA 数据安全公司的 MD5 消息摘要算法来计算字符串的 MD5 散列值。



虽然 explode()函数和 split()函数在功能上很相似，但应当使用 explode()函数而不是 split()函数，因为 split()函数要调用 PHP 的正则表达式解析引擎，速度上稍慢一些。

3. 基于数据库的身份认证

基于数据库的身份认证不仅改善了管理的方便性、提高了可扩展性，而且可以集成到更大的数据库基础设施中。基于数据库的身份认证使用所输入的用户名和密码作为查询条件，对用户表执行一个查询，如果可以从数据库表中查找到相关的记录，那么该用户通过数据库身份认证，否则给出相关的错误信息提示。下面来做一个实例，具体介绍基于数据库的身份认证。

在 MySQL 数据库中创建一个表，用来存储每个用户的用户名和密码等信息。

```
SET NAMES 'gbk';
use student_db;
CREATE TABLE userauth (
    ID TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,
    name VARCHAR(12) NOT NULL,
    username VARCHAR(10) NOT NULL,
    passwd VARCHAR(45) NOT NULL,
```

```
    PRIMARY KEY(ID)
) ENGINE=MyISAM DEFAULT CHARSET=gb2312;
```

接着，向新创建的表(userauth)中添加 3 条数据。

```
INSERT INTO userauth(name,username,passwd)
 VALUES("李大伟","user1","12345");
INSERT INTO userauth(name,username,passwd)
 VALUES("张莉莉","user2", "67890");
INSERT INTO userauth(name,username,passwd)
 VALUES('李武','user3', 'abcd123456');
```

将表和数据添加之后，编辑 PHP 代码，获取用户输入的用户名和密码来验证 userauth 表中存储的数据，代码如下：

```
<?php
    if(!isset($_SERVER['PHP_AUTH_USER']))
    {
        header('WWW-Authenticate: Basic realm="请输入用户名和密码"');
        header('HTTP/1.0 401 Unauthorized');
        exit;
    }
    else
    {
        //连接 MySQL 数据库，如果失败，输出"抱歉，不能连接到 MySQL 数据库!"
        mysql_pconnect("localhost","root","000") or die("抱歉，不能连接到 MySQL
数据库! ");
        //选择数据库
        mysql_select_db("student_db");
        //定义查询语句，获取用户输入的用户名和密码
        $query="select * from userauth where
username='".$_SERVER['PHP_AUTH_USER']."' and
passwd='".$_SERVER['PHP_AUTH_PW']."'";
        //执行查询的 SQL 语句
        $result=mysql_query($query);
        //获取受影响的函数，判断是否执行成功
        if(mysql_num_rows($result)==0)
        {
            header('WWW-Authenticate: Basic realm="请输入用户名和密码"');
            header('HTTP/1.0 401 Unauthorized');
            exit;
        }
        else
        {
            echo "欢迎您登录".$_SERVER['PHP_AUTH_USER'];
        }
    }
?>
```



上面的代码中多处用到 MySQL 相关的函数，可以参考后面的章节来对上面的代码做进一步了解。

在浏览器中运行该段代码，弹出一个对话框，如图 7-13 所示。当用户输入正确的用户名和密码时，关闭此对话框，并在页面中输出成功提示信息，如图 7-14 所示。

虽然 MySQL 认证功能较为强大，实际上实现起来是非常简单的。只要根据用户输入的用户名和密码对数据表执行一个查询，以用户输入的用户名和密码作为查询条件即可。这种基于数据库的身份认证不仅对 MySQL 数据库起作用，任何关系数据库都可以实现。

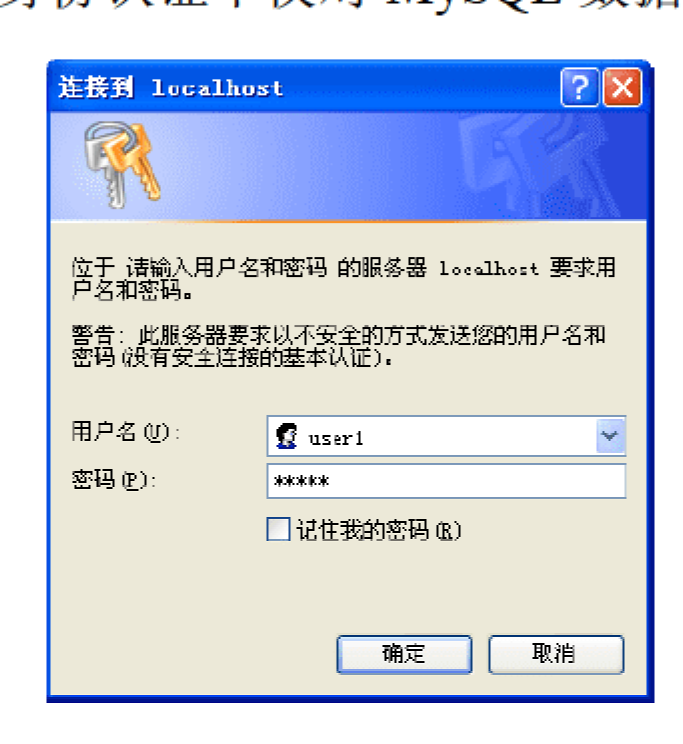


图 7-13　认证对话框

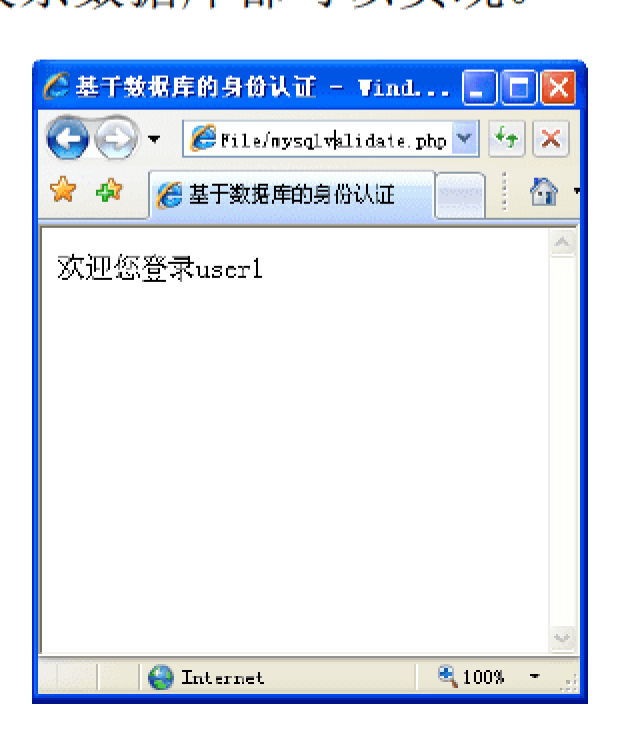


图 7-14　认证通过

4．基于 IP 的身份认证

在基于数据库的身份认证中，由于用户名和密码有可能会被黑客或一些别有用心的人窃取，所以为了解决这个问题，进一步保证用户及其信息的安全，一种有效的方法是当用户进行身份认证时，不仅需要合法的用户名和密码登录对，还需要一个特定的 IP 地址。下面就用一个实例来详细地介绍如何使用基于 IP 的身份认证验证用户信息的合法性。

修改前面创建好的 userauth 表，在该表中添加一个字段，命名为 ipaddress，并在该表中设置可以访问的 IP 地址。然后修改上面代码中的查询语句为：

```
$query="select * from userauth where username='".$_SERVER['PHP_AUTH_USER']."'
and passwd='".$_SERVER['PHP_AUTH_PW']."' and ipaddress='".$_SERVER
['REMOTE_ADDR']."'";
```

该段代码中的$_SERVER['REMOTE_ADDR']获取的是当前用户的 IP 地址，当用户是非法用户时，即使用户名和密码被恶意窃取，但是 IP 地址是无法改变的，即无法进行操作。

7.7.3　实例描述

当访问一个网站时，需要验证用户所输入的用户名和密码信息，判断用户是否为合法用户，这是 Web 开发中经常需要做的一件事。那么，下面就使用 PHP 中基于文件的身份认证方法来模拟这样的功能。

7.7.4　实例应用

【例 7-7】 使用基于文件的身份认证实现登录验证

(1)　新建 login.txt 文本文件，在该文件中输入以下内容并保存：

```
admin:admin
xianglin:xianglin
```

```
shaixingfu:xingfu
```

(2) 新建一个 PHP 文件(login.php)，该文件用于将用户输入的信息与 login.txt 文件中的登录对进行匹配，代码如下：

```
<?php
    $loginfile=file('login.txt');
    ||if(!(isset($_SERVER['PHP_AUTH_USER'])) !(isset($_SERVER['PHP_AUTH_PW'])))
    {
        header('WWW-Authenticate: Basic realm="请输入用户名和密码"');
        header("HTTP/1.0 401 Unauthorized");
        echo "本站提示：用户取消了登录！";
        exit;
    }
    else
    {
        foreach ($loginfile as $login)
        {
            list($username,$userpwd)=explode(':',$login);
            $userpwd=trim($userpwd);
            if(($username == $_SERVER['PHP_AUTH_USER']) && ($userpwd ==
$_SERVER['PHP_AUTH_PW']))
            {
                $authorized=true;
                echo $_SERVER['PHP_AUTH_USER']."身份认证成功,欢迎您!".$username;
                break;
            }
            else
            {
                //如果是错误的用户名或者密码，强制再次验证
                header('WWW-Authenticate: Basic realm="请重新输入用户名和密码
"');
                header("HTTP/1.0 401 Unauthorized");
                echo "本站提示：用户名或密码错误！";
                exit;
            }
        }
    }
?>
```

7.7.5　运行结果

打开 IE 浏览器，运行 login.php 页面，在页面中会显示登录窗口，如图 7-15 所示。在登录窗口中输入 login.txt 文件中的任一登录对，单击【确定】按钮，成功通过认证后将提示用户登录成功，如图 7-16 所示。如果用户单击了【取消】按钮，将会在 login.php 页面中，提示“用

户取消了登录”信息，或者用户重复输入几次错误的用户名或密码，将提示“用户名或密码错误”信息。

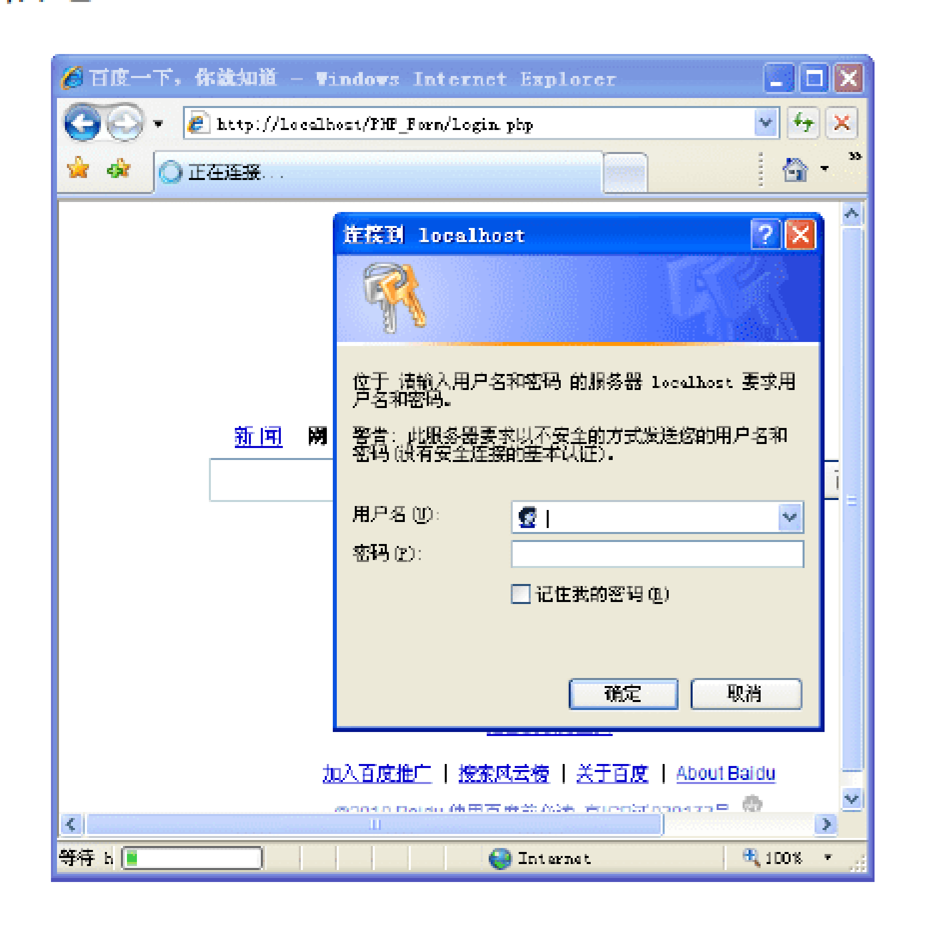


图 7-15　登录窗口



图 7-16　认证通过

7.7.6　实例分析



源码解析：

在上例中的 login.php 页面中，使用了 file()函数读取了 login.txt 文本文件的内容，它返回的是一个数组。在下面代码中，使用了 foreach 循环遍历了这个数组，并将用户输入的用户名和密码与数组中的元素进行一一的对比，来判断用户是否通过认证。

7.8　常见问题解答

7.8.1　PHP 和 JavaScript 交互的问题



PHP 和 JavaScript 交互的问题。

网络课堂：http://bbs.itzcn.com/thread-12475-1-1.html

知道 PHP 是运行在服务器端，JavaScript 是运行在客户端的，那么，如果我希望建个网页能远程查看我的服务器的运行情况，如 CPU 使用率、内存使用情况、网络使用情况，而且这些都要在网页中以图表的形式显示，希望显示的是动态实时变化的。这应该怎么做？希望说一下思路，提示下应该学哪种语言。

【解决办法】：完全的实时做不到，大概有 5 秒的间隔还是可以接受的吧？

思路：在前段网页上，用 JavaScript 使用 Ajax 每隔 5 秒向服务器发起一个请求查询当前服务器的系统状态，服务器端 PHP 处理这个请求，然后返回一个 JSON 或者 XML 的数据给前端，前端解析这个 JSON 或者 XML 数据，用 JavaScript 或者 Flash 展示出来。

后端用 PHP 实现，前端用一个 JavaScript 框架，例如用 ExtJS 的 Direct 和 Chart 组件，或者用 JQuery+Highchart 共同来解决。所以，要学习 JavaScript、HTML、PHP 语言。

7.8.2 PHP 与 JavaScript 之间变量引用问题



PHP 与 JavaScript 之间变量引用问题。
网络课堂：http://bbs.itzcn.com/thread-12476-1-1.html

PHP 与 JavaScript 交互，避免不了它们之间的变量引用问题，能总结一下它们之间变量引用的方法吗？

【解决办法】：首先来总结一下 JavaScript 变量传递到 PHP 脚本中的几种方法：

方法一：先用 PHP 生成 JavaScript 代码，然后通过浏览器的自动刷新，将 JavaScript 变量传递到 PHP 脚本中。注意：在实际应用中要防止用户在地址栏内更改传递的变量值。如：

```
<?php
if($qq) echo $qq.'<br>';
else echo "<script language='javascript'>var qq=' 我 的 QQ 是 12345678';
location.href='index.php?qq='+qq; </script>";
echo $qq;
?>
```

方法二：先用 document.cookie 方法把 JavaScript 变量写入 Cookie 中，再用 PHP 取出。

```
<script language="javascript" type="text/javascript">
var name = "myqq",value="10000";
document.cookie = name+" = "+value+";"
</script>
<?php
echo $_COOKIE["myqq"]."<br/>";
$my = "我的 QQ 是".$_COOKIE["myqq"];
echo $my;
?>
```

方法三：

```
<script>
var a = '我是旭日！';
</script>
<?php
$c = "<script>document.write(a);</script>";
echo $c;
?>
```

接着来总结一下 JavaScript 调用 PHP 变量的几种方法。

方法一：以 HTML 为媒介，遵循 PHP 变量->HTML 对象->JavaScript 调用。

```
<?php
$val_overview_pic = "我是旭日！";
?>
```

```
<input type="hidden" name="did" value="<?=$val_overview_pic;?>">
<script language="javascript" type="text/javascript">
alert(did.value);
//window.location ="/topic/isoktoshow.php?pic="+did.value;
</script>
```

方法二：用 PHP 语言编写 JavaScript 脚本。

```
<?php
$val_overview_pic = "/pic/pic1.jpg";
echo "<script language='javascript' type='text/javascript'>";
echo " alert('".$val_overview_pic."');";
//echo "                                window.location
='/topic/isoktoshow.php?pic='".$val_overview_pic ."';";
echo "</script>";
?>
```

7.8.3　POST 方法和 GET 方法的区别



POST 方法和 GET 方法的区别。
网络课堂：http://bbs.itzcn.com/thread-12477-1-1.html

表单的提交方式有两种，分别是 POST 和 GET，那么这两种提交方式有什么不同呢？它们的区别在哪里？哪种提交方式比较安全呢？

【解决办法】：它们的区别有以下几点。

第一点，POST 传输数据时，不需要在 URL 中显示出来，而 GET 方法要在 URL 中显示。POST 传输的数据量大，可以达到 2M，而 GET 方法由于受到 URL 长度的限制，只能传递大约 1024 字节。

第二点，POST 顾名思义，就是为了将数据传送到服务器端，GET 就是为了从服务器端取得数据，而 GET 之所以也能传送数据，只是用来告诉服务器，你到底需要什么样的数据。POST 的信息作为 HTTP 请求的内容，而 GET 是在 HTTP 头部传输的。

通过上面的叙述，可以总结如下：

① 使用 POST：数据有安全要求(包含密码)；数据量比较大(论坛里的发帖回帖)时，使用 POST 提交方式。

② 使用 GET：数据量少而且没有安全方面的顾虑的使用；某些不能用表单传值的时候(QueryString)时，使用 GET 提交方式。

7.8.4　PHP 表单数据接收的问题



JDK 安装错误 The download file appears to be corrupted. Please refer to the PHP 表单数据接收的问题。
网络课堂：http://bbs.itzcn.com/thread-12478-1-1.html

问一个问题，关于 PHP 里面的参数传输的问题，在 main.php 页面中用到表单，代码如下所示：

```
<form id="form1" name="form1" method="post" action="form4.php">
    <input name="name" type="text" size="18" maxlength="5" height="20" />
    <input type="submit" name="Submit" value="提交" />
    <input type="reset" name="Submit2" value="重置" />
    <input type="hidden" name=send1 value="send">
</form>
```

在 form4.php 页面中获取表单元素的值，代码如下：

```
<?php
    if($send1=="send")
    { echo("您的基本信息如下:");
    print("姓名:" .$name.);}
?>
```

这样编辑之后，却不能获取到名为“name”的文本框中的值，究竟要怎么样才能获取它的值呢？

【解决办法】：自 PHP 4.1 以后就取消了直接接收变量，而用$_GET[“变量名”]接收从链接传递过来的变量(或是表单传递过来的元素值)，用$_POST["变量名"]接收从表单提交过来的值。同样可以使用$_FILES[]、$_SERVER[]等。建议适应新的写法，或者把 php.ini 文件中的 register_globals 设为 on 也可以。

7.9 习　　题

一、填空题

(1) 在 PHP 页面中，要运行一段 PHP 代码，可以用以下语句声明：

```
<script language=________>
    脚本
</script>
```

画横线处应填__________。

(2) 将数据提交回表单所在脚本时，可以使用 PHP 超级全局变量__________。

(3) 如果表单中有多个复选框，PHP 获取选中的多个复选框的值的代码如下：

```
$_POST["shops"]
```

那么，需要将表单中的复选框元素的名称定义为__________。

二、选择题

(1) 请详细阅读下面列出的表单和 PHP 代码，当在表单里面的两个文本框中分别输入“php”和“great”的时候，PHP 将在页面中打印出__________。

```
<form action=index.php method="post">
<input type="text" name="element[]">
<input type="text" name="element[]">
    <input type="submit" value="提交" name="submit"/>
```

```
</form>
<?php
echo $_POST['element'];
?>
```

A. 什么都没有　　　　B. Array

C. php great　　　　D. great php

(2) 假设现在有一个 JavaScript 函数，如下面代码所示：

```
function validate_username($username)
```

在 PHP 代码中调用上述函数，并获取名称为“username”的表单元素的值作为 validate_username($username)函数的参数传递给该函数，则 PHP 代码为：__________。

A. validate_ username ($_POST['username'])

B. validate_ username ("$_POST['username'] ")

C. validate_ username ($username'))

D. validate_ username ("$username")

(3) 以下哪条 JavaScript 语句会产生运行错误：__________。

A. var obj=();　　　　B. var obj=[];

C. var obj={};　　　　D. var obj=//;

(4) 读取 POST 方法传递的表单元素值的方法是：__________。

A. $_post[“名称”]　　　　B. $_POST[“名称”]

C. $post[“名称”]　　　　D. $POST[“名称”]

三、上机练习

上机练习：使用 PHP 验证后台登录输入信息。

使用 PHP 语言生成后台登录页面中的验证码，如图 7-17 所示。当用户单击【登录】按钮时，查询数据库表中的信息，对用户的输入信息做合法验证。如果用户输入的验证码不正确，在验证失败提示信息页面输出“您输入的验证码不正确，请返回”，如图 7-18 所示；如果用户输入的用户名或者密码在数据库表中查询不到，则提示“你的用户名或者密码输入错误，请返回”，如图 7-19 所示；如果用户输入的用户名或者密码都合法，则提示“登录成功”，并跳转至后台主页面(index.htm)，如图 7-20 所示。



图 7-17　登录页面(PHP 自动生成验证码)



图 7-18　验证码输入不正确

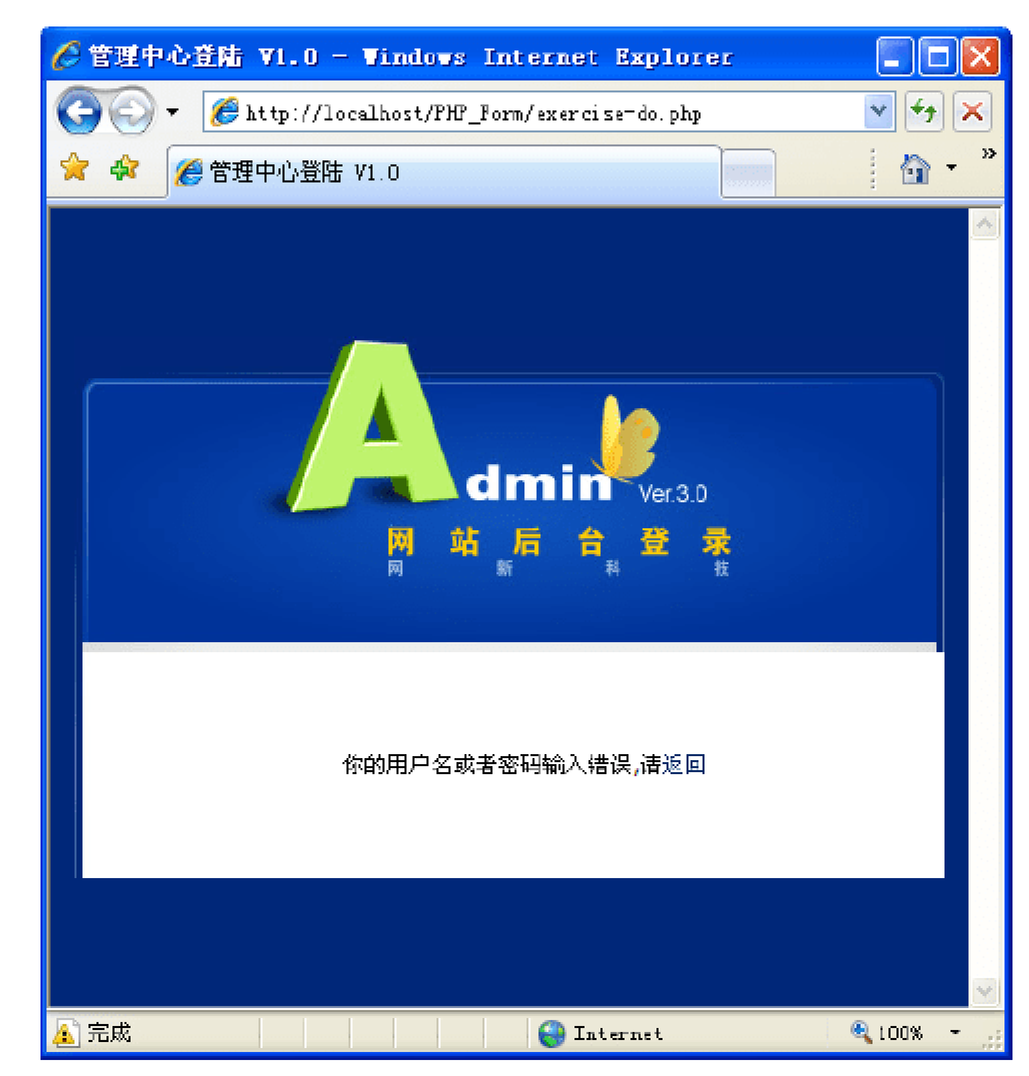


图 7-19　用户名或密码错误



图 7-20　登录成功



第 8 章　网页之间通信技术

内容摘要：

PHP 提供了一个行之有效的方法，管理用户的相关信息，以及用户的当前状态，当用户访问网站时，服务器将用户的有关信息(如首选项)存储在 Cookie 或 Session 中，并发送到浏览器，浏览器将保存这个 Cookie。用户再次访问该网站时，将从用户本地文件中读取相关信息。本章主要讲解 Cookie 和会话的相关知识，以实现网页之间信息的传递。例如如何创建 Cookie、删除 Cookie、创建会话、获取和设置会话 ID、使用会话存储数据等。

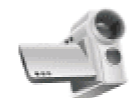
学习目标：

- 简要了解 Cookie。
- 掌握 Cookie 的基本操作。
- 熟练控制 Cookie 的有效性。
- 理解 PHP 会话的配置。
- 掌握会话的基本用法。
- 掌握传输会话 ID 及获得会话 ID。
- 掌握如何使用会话存储数据。
- 熟练使用 Session。

8.1 设置一个用户 Cookie

Cookie 技术是一个非常有争议的技术，自它诞生以来，就成为广大网络用户和 Web 开发人员的一个争论焦点。有一些网络用户，甚至包括一些资深的 Web 专家也对它的产生和推广感到不满，这倒不是因为 Cookie 技术的功能太弱或别的技术性能上的原因，而仅仅是因为他们觉得 Cookie 的使用对网络用户的隐私构成了危害。因为 Cookie 是由 Web 服务器保存在用户浏览器上的小文本文件，它包含有关用户的信息(如身份识别号码、密码、用户在 Web 站点上购物的方式或用户访问该站点的次数等等)。

那么 Cookie 技术究竟是怎样的呢？是否真的给网络用户带来了个人隐私的危害呢？还是让我们深入了解一下它再做回答吧。


视频教学：光盘\videos\08\cookie.avi　　　　长度：11 分钟

8.1.1 基础知识——Cookie 介绍

按照 Netscape 官方文档中的定义，Cookie 是在 HTTP 协议下，服务器或脚本维护客户工作站上信息的一种方式。无论何时用户链接到服务器，Web 站点都可以访问 Cookie 信息。

通俗地讲，浏览器用一个或多个限定的文件来支持 Cookie。这些文件在使用 Windows 操作系统的机器上叫做 Cookie 文件，在 Macintosh 机器上叫做 magic Cookie 文件，这些文件被网站用来存储 Cookie 数据。网站可以在这些 Cookie 文件中插入信息，这样对有些网络用户就有些副作用。有些用户认为这造成了对个人隐私的侵犯，更糟的是，有些人认为 Cookie 是对个人空间的侵占，而且会对用户的计算机带来安全性的危害。

目前有些 Cookie 是临时的，另一些则是持续的。临时的 Cookie 只在浏览器上保存规定的一段时间，一旦超过规定的时间该 Cookie 就会被系统清除。例如在 PHP 中 Cookie 被用来跟踪用户进程直到用户离开网站。持续的 Cookie 则保存在用户的 Cookie 文件中，下一次用户返回时，仍然可以对它进行调用。

技术文档	对于 Cookie 取舍的争论
在 Cookie 文件中保存 Cookie，一些用户会认为这将带来很大的问题。主要是有些用户担心 Cookie 会跟踪用户网上冲浪的习惯，譬如用户喜爱到哪些类型的站点、喜欢从事一些什么活动等。害怕这种个人信息一旦落入一些别有用心的人手中，那么用户就可能成为一大堆垃圾广告的受扰者，甚至遭受意外的损失。不过，这种担心压根儿不会发生，因为网站以外的用户是无法跨过网站来获得 Cookie 信息的。所以想以这种目的来应用 Cookie 是不可能的。不过，由于一些用户错误的理解以及“以讹传讹”，一些浏览器开发商别无选择，只好作出相似的回应，例如 Netscape Navigator 4.0 和 Internet Explorer 3.0 都提供了屏蔽 Cookie 的选项。 对 Cookie 技术期待了这么久的结果是，迫使许多浏览器开发商在它们的浏览器中提供了对 Cookie 的灵活控制功能。例如，目前的两大主流浏览器 Netscape Navigator 和 Internet Explorer 是这样处理 Cookie 的：Netscape Navigator 4.0 不但可以接受 Cookie 进行警告，而且还可以屏蔽掉 Cookie；InternetExplorer 3.0 也可以屏蔽 Cookie；在 Internet Explorer 4.0 中就只能进行接受警告而没有提供屏蔽	

选项，不过在 Internet Explorer 4.0 之后的更新版本中又加入了屏蔽 Cookie 的功能选项。

此外，很多最新的技术甚至已经可以在不能屏蔽 Cookie 的浏览器上进行 Cookie 的屏蔽了。例如，可以通过将 Cookie 文件设置成不同的类型来限制 Cookie 的使用。但是，非常不幸的是，要是你想完全屏蔽 Cookie 的话，肯定会因此拒绝许多的站点页面。因为现在已经有许多 Web 站点开发人员爱上了 Cookie 技术的强大功能，例如 Session 对象的使用就离不开 Cookie 的支持。

尽管今天仍有一些网络用户对 Cookie 有争论，但是对于绝大多数的网络用户来说还是倾向于接受 Cookie 的。因此，现阶段我们尽可以放心地使用 Cookie 技术来开发我们的 Web 页面。

8.1.2　基础知识——Cookie 的工作原理

要了解 Cookie，就要知道它的工作原理。一般来说，Cookie 通过 HTTP Headers 从服务器端返回到浏览器上。首先，服务器端在响应中利用 Set-Cookie Header 来创建一个 Cookie，然后浏览器在它的请求中通过 Cookie Header 包含这个已经创建的 Cookie，并将它返回至服务器，从而完成浏览器的验证。

例如，我们创建了一个名字为 login 的 Cookie 来包含访问者的信息，创建 Cookie 时，服务器端的 Header 如下面所示，这里假设访问者的注册名是“Michael Jordan”，同时还对所创建的 Cookie 的属性，如 path、domain、expires 等做了指定。

```
Set-Cookie:login=Michael Jordan;path=/;domain=msn.com;
expires=Monday,01-Mar-99 00:00:01 GMT
```

上面这个 Header 会自动在浏览器端计算机的 Cookie 文件中添加一条记录。浏览器将变量名为“login”的 Cookie 赋值为“Michael Jordon”。注意，在实际传递过程中这个 Cookie 的值是经过 URLEncode 方法的 URL 编码操作的。

这个含有 Cookie 值的 HTTP Header 被保存到浏览器的 Cookie 文件后，Header 就通知浏览器将 Cookie 通过请求以忽略路径的方式返回到服务器，完成浏览器的认证操作。



浏览器创建了一个 Cookie 后，对于每一个针对该网站的请求，都会在 Header 中带着这个 Cookie；不过，对于其他网站的请求 Cookie 是绝对不会跟着发送的。而且浏览器会这样一直发送，直到 Cookie 过期为止。

客户端的每一项数据都是相互独立的，在需要读取时再分别取出。Cookie 就是将数据存放在客户端的技术之一，使用过程如图 8-1 所示。

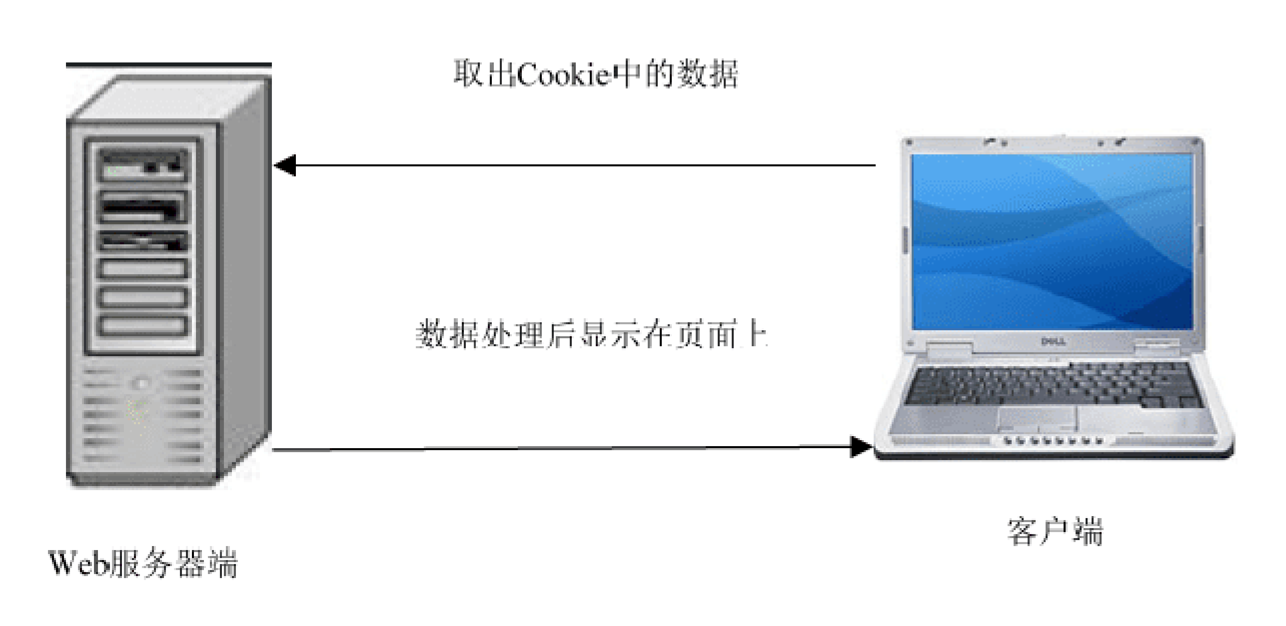


图 8-1　Cookie 的使用过程

8.1.3 基础知识——Cookie 的操作

Cookie 在许多方面都是很有用的，例如，许多 Web 站点都使用 Cookie 来存储用户的身份和查看首选项。当用户返回该 Web 站点时，Cookie 使浏览器能够识别这个用户并恢复该用户选定的站点选项。可见 Cookie 在会话中起到了很重要的作用，本节将接受如何设置创建 Cookie，以及一些对 Cookie 的基本操作。

1. 创建 Cookie

在 PHP 中创建 Cookie 很简单，只需调用 setCookie()函数即可，该函数的语法格式如下所示：

```
bool setcookie(string name[,string value [,int expire [,string path [,string
domain[,int secure]]]]])
```

setCookie()函数的上述参数除了 name 参数之外，其他参数都是可选项。name 参数指定 Cookie 的名称，value 参数指定 Cookie 值。expire 参数是一个有 time()函数或者 mktime()函数返回的 Unix 格式的函数，用于指定 Cookie 的过期时间，过了这个时间，将无法访问 Cookie。secure 参数表示 Cookie 只在建立 HTTP 连接时才发送。



Cookie 创建完成之后不会立即生效，而是等到请求下一个页面时才能生效。这是因为设置好的 Cookie 会由服务器端传递给客户端，在请求下一个页面时，客户端才能将 Cookie 取出并传回服务器端。

下面创建几个 Cookie 试试，代码如下所示：

```
setcookie("username","Jack");                    //简单创建一个 cookie
setcookie("password","123456",time()+7200);      //创建一个带有有效时间为两个小时
(7200 秒)的 cookie
setcookie("cookie","cookie value",time+7200,"/forum",".phpuser.com",1);
                                                 //创建一个完整的 cookie
```

也可以通过在 Cookie 名称中使用数组符号来设定数组 Cookie，可以设定多个 Cookie 作为数组单元，在脚本提取 Cookie 时所有的值都放在一个数组中，代码如下所示：

```
                                                 // 设定 cookie
setcookie("cookie[]","cookiethree");
setcookie("cookie[]", "cookietwo");
setcookie("cookie[]", "cookieone");
```

2. 接受和处理 Cookie

由于 PHP 可以自动将创建好的 Cookie 值作为一个 PHP 变量使用，所以使用 PHP 接受和处理 Cookie 非常容易。该变量名称与 Cookie 名称相同。下面通过一个例子具体演示说明在 PHP 中如何接受和处理 Cookie，代码如下所示：

```
<?php
    // 设定 cookie
    setcookie("cookie[0]","cookieone",time()+3600);
```

```
    setcookie("cookie[1]", "cookietwo",time()+3600);
    setcookie("cookie[2]", "cookiethree",time()+3600);
    // 刷新页面后,显示出来
    echo $_COOKIE["cookie"]."<br/>";
    print_r($_COOKIE);
?>
```

上述代码执行结果如下所示：

```
Array
Array (
    [cookie] => Array (
        [0] => cookieone
        [1] => cookietwo
        [2] => cookiethree
    )
)
```

3．删除 Cookie

前面已经提到过使用 time()函数可以设置一个 Cookie 的有效时间，当超过这个时间时，将被自动删除，不过用户也可以立即删除一个 Cookie。PHP 删除一个已经存在的 Cookie 有两个方法：

(1) 调用只带有 name 参数的 setCookie()，那么名为 name 的 Cookie 将被从用户计算机上删除。

(2) 设置 Cookie 的失效时间为 time()或者 time()-1，那么这个 Cookie 在这个页面浏览完毕之后就被删除了，其实是失效了。

下面删除一个 username，代码如下所示：

```
setcookie("username");
setcookie("username",time()-12);
```

下面看删除用户名与密码的例子，代码所示：

```
<pre>
<?php
    setcookie("username","hello",time()+7200);
    setcookie("password","123456",time()+7200);
    echo $_COOKIE["username"]."\n";
    echo $_COOKIE["password"]."\n";
    setcookie("username");
    setcookie("password","",time()-1800);
    print_r($_COOKIE);
?>
</pre>
```

8.1.4　实例描述

通过上面的学习是不是对 Cookie 有所了解了呢？使用 Cookie，网站对用户的管理是不是

方便很多呢？但具体如何为用户设置一个 Cookie 呢？本节实例将具体讲解如何设置一个用户 Cookie。

8.1.5 实例应用

【例 8-1】 设置一个用户 Cookie

(1) 创建一个 PHP_10 项目，在该项目文件夹下再新建一个 login_frm.php 文件作为用户登录表单页面，代码如下所示：

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;charset=utf8" />
    </head>
    <body>
        <form method="post" action="login.php">
            用户名:<input type="text" name="username" /><br/>
            密码:<input type="password" name="password"/><br/>
            <input type="submit" value="提交" />
        </form>
    </body>
</html>
```

(2) 在项目目录下新建一个 login.php 文件，用于接收用户输入的 username，并将 username 设置到 Cookie 中，代码如下所示：

```
<?php
    if($_POST['username'] = 'admin')
    {
        setcookie('haha','admin');
        header("location:index.php");
    }
?>
```

(3) 在项目目录下新建一个 Cookie.php 文件，在该文件中获取 Cookie，并输出显示用户是否设置了 Cookie。代码如下所示：

```
<?php
    if($_COOKIE['haha'] == 'admin')
    {
        echo $_COOKIE['haha'];
        echo'你设置了 cookie';
    }
    else
    {
        echo'你没有设置 cookie';
    }
?>
```

8.1.6　运行结果

项目编写完成之后，打开 IE 浏览器，在地址栏中输入“http://localhost/PHP_10/login_frm.php”，即可进入登录页面，执行效果如图 8-2 所示。



图 8-2　用户登录页面

用户输入用户名和密码之后，单击【提交】按钮，可以看到提示 Cookie 已经设置成功了，执行效果如图 8-3 所示。

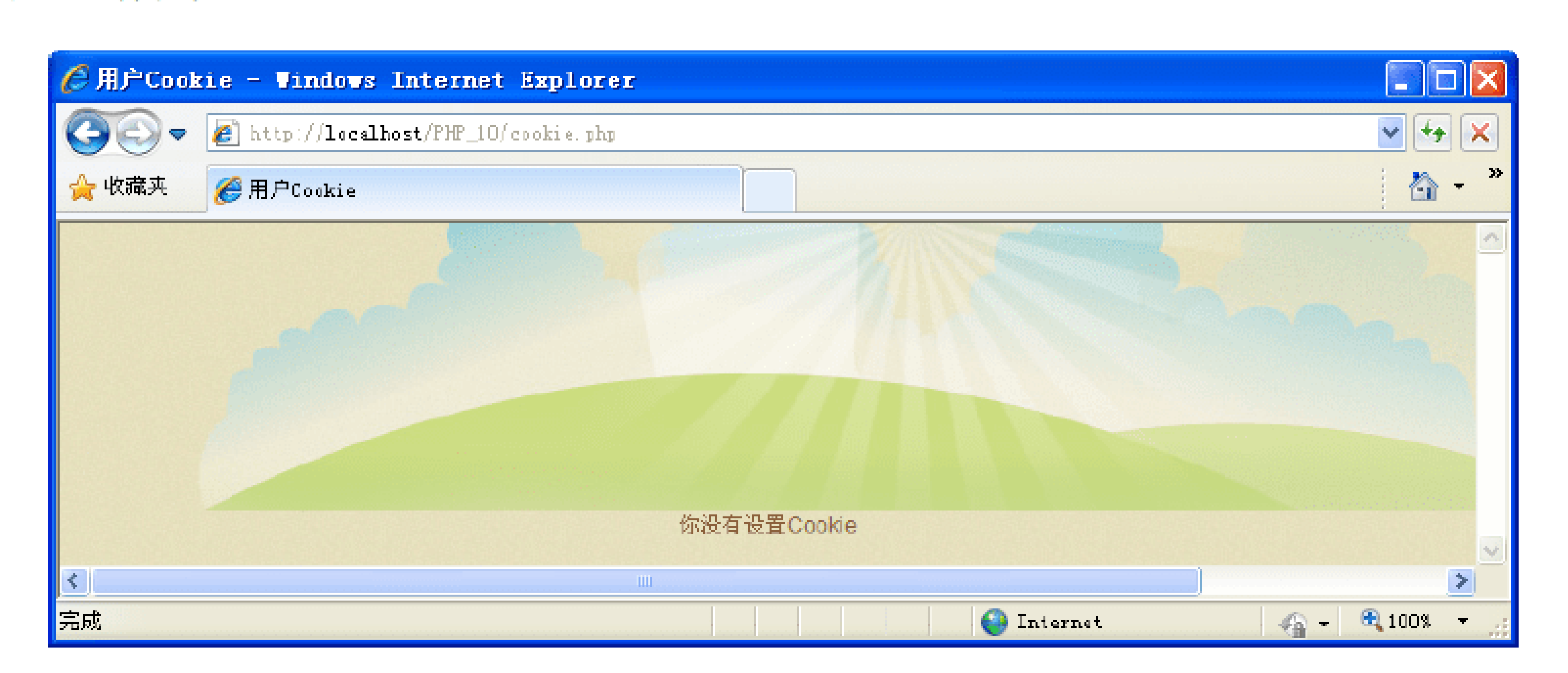


图 8-3　用户 Cookie 设置成功

8.1.7　实例分析

源码解析：

本实例首先定义了一个用户登录页面 login_frm.php，用户可以输入用户名和密码进行登录。然后创建了一个 login.php 页面，该页面中获取用户输入的用户名，并将用户名设置到 Cookie 中。最后在 Cookie.php 页面中，获取创建的 Cookie，显示用户是否创建了 Cookie。

8.2 Cookie 有效性控制

通过上面的学习，对 Cookie 的创建、设置、删除操作应该掌握得差不多了，下面我们来学习如何设置 Cookie 的有效期，Cookie 保存在客户端，设置有效期限可以增强网络用户的信息安全。

视频教学：光盘\videos\08\cookie.avi　　长度：11 分钟

8.2.1 实例描述

现在有很多用户排斥浏览器支持 Cookie 记录，认为这样会泄露他们的隐私，比如个人兴趣爱好、生活习惯等。一旦这些信息被一些别有用心的人得到，会造成很大麻烦，甚至有生命危险。但如果浏览器真的不支持 Cookie 的话，网站对用户的管理又会很困难。这是不是很矛盾呢？实际上有一个合理的解决办法，那就是设置 Cookie 的有效性，即设置 Cookie 的有效时间，过了这个时间，Cookie 将会被自动销毁。本节实例就来具体演示如何来设置 Cookie 的有效性。

8.2.2 实例应用

【例 8-2】 设置用户登录 Cookie 有效期

(1) 在 PHP_10 项目目录下新建一个 user_login.html 页面，用户可以通过该页面输入用户登录信息，并设置登录 Cookie 的有效时间，代码如下所示：

```
<html>
    <head>
        <title>Login</title>
        <meta http-equiv="Content-Type" content="text/html; charset=gb2312">
    </head>
    <body>
        <form name="form1" method="post" action="usercookie.php">
            <table  width="300"  border="0"  align="center"  cellpadding="2"
cellspacing="2">
                <tr>
                    <td width="150"><div align="right">用户名：</div></td>
                    <td width="150"><input type="text" name="username"></td>
                </tr>
                <tr>
                    <td><div align="right">密码：</div></td>
                    <td><input type="password" name="passcode"></td>
                </tr>
                <tr>
                    <td><div align="right">Cookie 保存时间：</div></td>
```

```
                    <td><select name="savetime" id="savetime">
                    <option value="60">一分钟</option>
                    <option value="1800" selected>半个小时</option>
                    <option value="3600">一个小时</option>
                    </select></td>
                </tr>
            </table>
            <p align="center">
                <input type="submit" name="Submit" value="登录">
                <input type="reset" name="Reset" value="重置">
            </p>
        </form>
    </body>
</html>
```

(2) 新建一个 userCookie.php 页面，在该页面中获取用户输入的用户名和密码，并使用它们设置两个 Cookie，分别是 username 和 passcode，在设置这两个 Cookie 的有效时间，代码如下所示：

```
<html>
<head>
<meta http-equiv="content-type" content="text/html;charset=gb2312" />
<title>设置 cookie 的有效期</title>
</head>
<center>
<?php
    $username=$_POST['username'];
    $passcode=$_POST['passcode'];
    $savetime = $_POST['savetime'];
    setcookie("usernamecookie",$username,time()+$savetime);

?>

<?php
    if($username==""or $passcode=="")
    {
        echo "请输入用户名称或密码";

    }else{
        echo "cookie 数据已经顺利存储<hr/><p>";
        echo "目前\$usernamecookie 是:【".$username;
        echo "】有效期是: ".$savetime."秒</p>";

    }
    echo "<br/><a href='user_login.html'>重新输入</a>";

?>
</center>
<body></body>
</html>
```

8.2.3 运行结果

打开 IE 浏览器，在地址栏中输入“http://localhost/PHP_10/user_login.html”，进入用户登录页面，执行效果如图 8-4 所示。

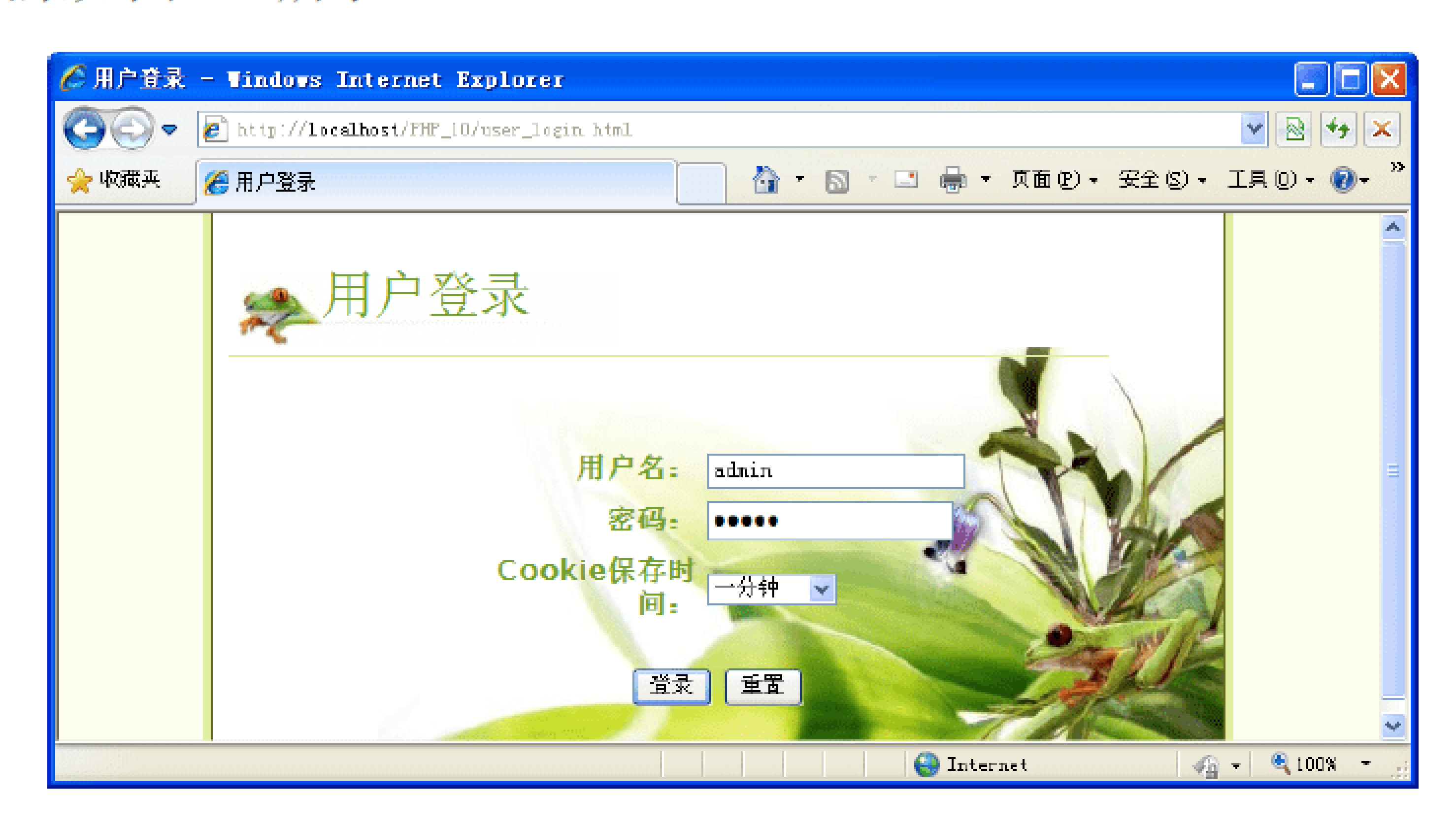


图 8-4 用户登录

当用户输入登录信息，选择 Cookie 保存时间之后，单击【登录】按钮，可以进入设置 Cookie 有效期页面，执行效果如图 8-5 所示。



图 8-5 设置 Cookie 的有效期

8.2.4 实例分析


源码解析：

本实例首先创建了一个用户登录页面 user_login.html，包含了一个用户登录表单，可以输入用户名、密码和选择 Cookie 保存时间。当用户输入登录信息完毕之后，可以单击【登录】按钮提交表单到 savetime.php 页面，在该页面中设置 Cookie 的有效时间。最后在页面中输出 Cookie 的有效时间。

8.3　使用 Session 实现购物车

使用 PHP 中的会话可以使用户建立更灵活的应用并提高网站的吸引力。本节将具体介绍 PHP 的会话机制、会话配置以及基本用法。

 视频教学：光盘\videos\08\配置 Session 会话.avi　　长度：3 分钟

8.3.1　基础知识——认识回话

会话(Session)是一种为特定用户识别和管理状态的方式，当会话发生时，每位网站访问者都会得到一个会话 ID(SID)，并将此会话 ID 与任意数量的数据关联。服务器可以使用会话 ID 在处理请求之前定位相应的会话。



比起 Cookie，Session 是存储在服务器端的会话，相对安全，并且不像 Cookie 那样有存储长度限制。

由于 Session 是以文本文件形式存储在服务器端的，所以不担心客户端修改 Session 内容。实际上在服务器端的 Session 文件，PHP 自动修改 Session 文件的权限，只保留了系统读和写权限，而且不能通过 FTP 修改，所以安全得多。

技术文档	对比了解 Session 的优势
对于 Cookie 来说，假设我们要验证用户是否登录，就必须在 Cookie 中保存用户名和密码(可能是 md5 加密后的字符串)，并在每次请求页面的时候进行验证。如果用户名和密码存储在数据库，每次都要执行一次数据库查询，给数据库造成多余的负担。因为我们并不能只做一次验证。为什么呢？因为客户端 Cookie 中的信息是有可能被修改的。假如你存储$flag 变量来表示用户是否登录，$flag 为 true 的时候表示登录，为 false 的时候表示未登录，在第一次通过验证后将$flag 等于 true 存储在 Cookie，下次就不用验证了，这样对吗？错了，假如有人伪造一个值为 true 的$flag 变量那不是就立即取得了管理权限吗？非常不安全。 而 Session 就不同了，Session 是存储在服务器端的，远程用户没办法修改 Session 文件的内容，因此我们可以单纯存储一个$flag 变量来判断是否登录，首次验证通过后设置$flag 值为 true，以后判断该值是否为 true，假如不是，转入登录界面，这样就可以减少很多数据库操作了。而且可以减少每次为了验证 Cookie 而传递密码的不安全性了(Session 验证只需要传递一次，假如你没有使用 SSL 安全协议的话)。即使密码进行了 md5 加密，也是很容易被截获的。	

当然使用 Session 还有很多优点，例如控制容易，可以按照用户自定义存储(存储于数据库)等。这里就不再多讲了。

8.3.2　基础知识——PHP 会话配置

在 PHP 中有许多配置选项负责确定 PHP 会话处理功能的行为，这些配置选项在确定这种

行为时会起到很重要的作用。配置选项的形式是“session.*”，在 php.ini 文件中可以看到这些选项。本节将介绍比较重要的配置选项。

1. session.save_handler(files,mm,sqlite,user)

该选项定义用来存储和获取 Session 数据的处理器的名字，默认值为 files，也就是使用文件来存储 Session，还可以使用共享内存(mm)、SQLite 数据库(sqlite)和用户自定义函数(user)。这四个配置项的具体意义如下。

- file：会话存储文件的数量可能会达到几千个，甚至几十万个。
- 共享内存：速度最快，但却最不可靠，因为数据存储在 RAM 中。
- sqlite：利用了新的 SQLite 扩展，通过这个轻量级数据库同名的管理会话信息。
- 用户自定义：虽然配置复杂，但也是最灵活、功能最强大的一个选项，因为可以创建定制处理函数，在开发人员所需的任何媒体上存储信息。



在 windows 系统下，必须改变该选项才能使用 Session 相关函数，不过使用函数 session_set_save_handler()可以在网页中改变这个配置。

2. session.save_path(string)

定义了传递给处理器的参数，如果选择了默认的 files 文件处理器，则此值是创建的文件的路径，在 Unix 上默认的/tmp，在 windows 上，默认的是 C:\windows\temp。使用函数 session_save_path()可以改变这个配置。

该指令还有一个可选的参数 N 来决定会话文件分布的目录深度。例如，设定为“5;c:\windows\temp”将使创建的会话文件和路径类似于 c:\windows\temp\4\b\1\4\3\sess_4b143adadlsfjlsdkjfldsjfkf，要使用 N 参数，必须在使用前先创建好这些目录。

使用该配置项还要注意以下 3 点：

(1) 如果使用了 N 参数并且 N 大于 0，那么将不会执行自动垃圾回收。

(2) 如果使用了 N 参数，那么要确保 session.save_path 的值用双引号括起来，因为分隔符分号在 php.ini 中也是注释符号。

(3) 该配置选项不能设置为位于服务器文档根目录下的某个目录，否则服务器上的其他用有可能通过该目录的文件列表破解会话。



在 PHP 4.3.6 之前，windows 用户必须修改此选项以激活 PHP 会话函数，并且必须制定一个合法的路径，例如：C:\temph 或 C:\windows\temp。

3. session.name(string)

该选项用于指定 Cookie 名称，它只能由字母和数字组成，其作用域为 PHP_INI_ALL，默认值为 PHPSESSID。这里可以将默认值改为更适合于应用程序的名字，或者可以通过 session_name()函数在需要时进行修改。

4. session.auto_start(boolean)

该选项用于指定会话模块是否在请求开始时自动启动一个会话，其作用域为 PHP_INI_ALL，默认是 0，表示不启动。如果将选项设置为 1，则自动启动会话。

自动启动会话可以维护由单个客户端发布的一组请求信息，每个网页请求都会启动创建会话，如果网页不依赖会话信息，那么就应该关闭自动启动，这样可以减少 PHP 应用程序服务器处理脚本的时间，从而提高效率。如果特定的网页需要提供会话支持，可以在网页内调用 session_start()函数启动它。



启动该指令将无法在会话中存储对象，因为类定义要在会话开始之前加载才能重新创建对象。但 session.auto_start 会禁止这一点，所以如果要在会话中管理对象，就要禁用该指令。

5．session.serialize_handler(string)

定义用来序列化或逆序列化的处理器名字，当前支持 PHP 内部格式(名为 php)和 WDDX(名为 wddx)，默认为 php，但是，如果 PHP 编译时加入了 wddx 支持，则只能用 wddx。

6．session.gc_probability(integer)

该配置选项和 session.gc_divisor 选项配合使用，可以管理 gc(Garbage Collection，垃圾回收)进程启动的概率，默认为 1。

7．session.gc_divisor(integer)

与 session.gc_probability 项配合起来使用，定义了在每个会话初始化时启动 gc 进程的概率。此概率使用 session.gc_probability/session.gc_divisor 计算获得，例如 1/100 意味着在每个请求中有 1%的概率启动 gc 进程。session.gc_divisor 默认为 100。

8．session.gc_maxlifetime(integer)

该配置选项指定过多少秒后数据就会被视为“垃圾”，将会执行 gc，从而被清除。其作用域为 PHP_INI_ALL。默认值 1440(默认 24 分钟，即 1440 秒)。

9．session.referer_check(string)

该配置选项用于指定一个用于验证每位用户的子串，如果没有包含此子串，会话 ID 将失效。该子串默认为空字符串。如果使用 URL 重写来传播会话 ID，很多人都能通过复制和散布 URL 来查看某个会话状态，使用该指令可以减少这种可能性。

10．session.entropy_file(string)

该配置选项用于给出一个到外部资源(文件)的路径，并将这个资源集成到会话 ID 生成过程中。在 Windows 系统下，安装 Cygwin(http://www.cygwin.com/)会提供类似于 random 或 urandom 的功能。而在 Unix 系统下，这个来源通常是/dev/random 或/dev/urandom。

11．session.entroy_length(integer)

该配置选项确定从 session.entropy_file 选项所指定文件中读取的字节数。如果 session.entropy_file 为空，则此选项被忽略，并使用标准的会话 ID 生成模式。

12．session.use_cookies(boolean)

该配置选项用于指定是否在客户端用 Cookie 来存放会话 ID。设置为 1，表示使用 Cookie 进行会话 ID 传播；设置为 0，将使用 URL 重写。其作用域为 PHP_INI_ALL，默认值是 1(表

示启用)。如果要在多次访问网站之间维护用户会话，就应当使用 Cookie，这样处理器可以重新得到会话 ID，继续原来所保存的会话。如果用户数据只用于一次访问，就可以使用 URL 重写。

13. session.use_only_cookies(boolean)

该配置选项指定是否在客户端仅使用 Cookie 来存放会话 ID，启用此设置可以防止别有用心者通过 URL 传递会话 ID 的攻击。其作用域为 PHP_INI_ALL；默认值是 0(表示同时使用 Cookie 和 URL 重写)，设置为 1，表示 PHP 只使用 Cookie。

14. session.cookie_lifetime(integer)

该配置选项以秒为单位，指定发送到浏览器的 Cookie 的有效期，其作用域为 PHP_INI_ALL，默认值是 0，表示“Cookie 将存活到浏览器关闭”。如果要使 Cookie 存活 1 小时，就将该选项设置为 3600。

15. session.cookie_path(string)

该配置选项用来指定要使用的会话 Cookie 的路径。对于有效路径下的所有子目录，此 Cookie 同样有效。例如，如果该选项设置为“/”，则 Cookie 对整个网站都有效。如果设置为“/bbs”，则只有在 http://www.itzcn.com/bbs/路径中调用 Cookie 才有效。

16. session.cookie_domain(string)

该配置选项指定要使用的 Cookie 的域名。默认值为空，表示将根据 Cookie 规范产生 Cookie 的主机名。该选项是必须的，因为它能防止其他域读取用户的 Cookie。该选项的设置如下所示：

```
session.cookie_domain = www.itzcn.com;
```

如果要在网站子域中使用会话，假如现在有 bbs.itzcn.com、books.itzcn.com 和 download.itzcn.com 子域，该选项可以如下所示设置：

```
session.cookie_domain = .itzcn.com;
```

17. session.cookie_secure(boolean)

该参数设置 Cookie 的安全标记，这样可以防止浏览器通过不加密的连接发送会话 Cookie。当该参数设置为 on 时，浏览器通过一个使用 SSL(安全套接字层)保存的网络连接发送会话 Cookie。其作用域为 PHP_INI_ALL，默认值是 off，表示允许浏览器通过加密和不加密的服务发送会话 Cookie。

18. session.cache_limiter(string)

该配置选项用来指定会话页面使用的缓存控制方法，默认为 nocache。在页面中也可以使用 session_cache_limiter()函数控制。它有 5 个可选值如下所示。

- nono：该设置禁止随启用会话的页面传输任何缓存控制首部。
- nocache：该设置确保对于每个请求，在可能提供缓存的版本之前，先将请求发送到最初的服务器。
- private：该设置指定缓存的文档是私有的，该文档只用于最初的用户，不能与其他用户共享。

- private_no_expire：该设置是 private 的变体，它保证不会向浏览器发送任何文档过期日期。
- public：该设置认为所有文档都是可缓存的，最初的文档请求可能需要认证。

19．session.cache_expire(integer)

该配置选项确定在创建新页面之前缓存的会话页面可用的时间(以秒为单位)。其作用域为 PHP_INI_ALL。默认值是 180。如果 session.cache_limiter 设置为 nocache，则该指令将被忽略。

20．session_use_trans_sid(boolean)

该配置选项被启用后，可以消除使用 URL 重写过程中发生错误的可能性。其作用域为 PHP_INI_SYSTEM | PHP_INI_PERDIR。默认值是 0(表示禁用)。如果禁用 session.use_Cookies，为了确保传播 ID，用户的会话 ID 必须增加到 URL。用户可以手工将变量$SID 追加到每个 URL 的后面，也可以通过启用指令自动处理。

如果浏览器关闭了 Cookie 支持，想用 Session，就必须激活此选项：

```
session.use_trans_sid=1
```

21．session.hash_function(integer)

允许用户指定生成会话 ID 的散列算法。可选值如下所示：

- 0　MD5 加密(128 位)
- 1　SHA-1 加密(160 位)

22．session.hash_bits_per_character

允许用户定义在将二进制散列数据转换为可读的格式时每个字符存放多少个比特。可选值包括 4、5、6，如下所示：

- 4 bits　0-9，a-r
- 5 bits　0-9，a-v
- 6 bits　0-9，a-z,A-z,"-",","

23．url_rewriter.tags(string)

该指令的作用域为 PHP_INI_ALL，默认值是 a=href,area=href,frame=src,input=src, form=fakee-ntry。启用 session.use_trans_sid 时，在将文档发送给客户端之前，会话 ID 会自动追加到所请求文档中的 HTML 标记后面。但是，与超链接或表单标记不同，许多标记在启动服务器请求时不起作用，所以可以使用 url_rewriter.tags 指令告诉服务器会话 ID 应当追加到哪些标记后面。



如果要符合 XHTML，去掉 form 项并在表单字段前后加上<fieldset>标签。

8.3.3　基础知识——基本用法

PHP 会话可以保存在文件中，会话文件一般都很小，但是文件数目却很多。本节将具体介

绍会话的一些基本用法，如开始会话、创建会话变量及删除撤销会话。

1. 开始会话

PHP 4.0 有一个一直为人所青睐的特性，就是 PHP 的会话(Session)支持。相比之下，PHP 3.0 的用户则不得不使用第三方的程序库或完全不能具备这项功能。缺乏会话支持能力是 PHP 最大的缺陷之一，也是它最受人指责的地方。不过，随着会话支持从早期测试版本的 PHP 4.0 开始，这个障碍也荡然无存了。

有了会话支持，我们就可以在用户访问网络站点期间维持用户特定的变量而无须设置多个 Cookie、使用隐蔽表单域等。

在 PHP 中，可以通过使用 session_start()函数来实现记忆和启动恢复会话功能，该函数的语法格式如下所示：

```
bool session_start(void)
```

使用 session_start()函数创建一个新会话或者继续当前会话，这取决于是否拥有会话 ID。开始会话时，只需要调用 session_start()函数：

```
session_start();
```



使用 session_start()函数，添加任何异常处理都不起作用，因为无论结果如何，session_start()函数都会报告成功的结果。如果启用了配置选项 session_auto_start，不必执行该函数，就可以开始会话，但是这样一来每次启用 PHP 页面都会开始或继续一个会话。

2. 创建会话变量

在使用 PHP 4.0 之前，开发人员习惯使用 session_register()函数创建会话变量。在一个会话初始化之前，调用 session_start()函数打开会话文件，并且通过 session_register()函数创建会话变量，如下所示：

```
session_register("build");
$build = "http://www.itzcn.cn";
```

上述代码中变量$build 与其值被自动保存在会话存储器中，这里应该注意，将变量的名称而不是变量本身传递给 session_register()函数。会话变量一旦创建，就变得持久并对初始化会话的脚本可用。

另外，千万不要在脚本结束之前显示保存会话变量，因为 PHP 会追踪会话变量的值并将值保存到会话文件。

但在 PHP 4.0 之后，推出了另外一种方便的方法，也是现在创建会话变量的首选方法，即像创建其他任何变量一样设置，不过需要使用超级全局变量$_SESSION 在上下文中引用这些变量。假如现在需要创建一个名为 customer 的会话变量，具体代码如下所示：

```
<?php
    session_start();
    $_SESSION['customer'] = "beautiful girl";
    echo "Welcome ".$_SESSION['customer'];
?>
```

执行上述代码，将在页面中输出如下信息。

```
Welcome beautiful girl
```

3. 删除会话变量

学习了创建会话变量，大家一定会马上联想到，能否删除会话变量呢？答案当然是肯定的。PHP 提供了 unset()函数可以删除会话变量。下面就来看看如何实现删除会话变量，代码如下所示：

```
<?php
    session_start();
    $_SESSION['customer'] = "beautiful girl";
    echo "Welcome ".$_SESSION['customer']."<br/>";
    unset($_SESSION['admin']);
    echo "The Customer is:".$_SESSION['customer']."<br/>";
?>
```

执行上述代码，在页面中将输出如下信息：

```
Welcome beautiful girl
The Customer is:
```

从上述执行结果可以看出，customer 会话变量已经被删除掉了。

4. 撤销会话

撤销会话可以通过配置 PHP 的会话处理选项，根据过期时间或概率自动销毁会话，但有时手工撤销会话也需要使用手工消除当前会话。例如用户在离开网站时注销登录，或者当用户单击适当的链接时，可以从内存中消除当前会话中的所有变量，甚至从存储机制中完全消除整个会话，从而实现手工破坏会话，这些功能可以通过使用 PHP 提供的 session_unset()函数和 session_destroy()函数完成。本节介绍这两个函数的具体使用。

1) session_unset()函数

该函数的使用语法格式如下所示：

```
void session_unset();
```

session_unset()函数可以用于清除存储在当前会话中的所有变量，能有效地将会话重置为创建时的状态(没有注册任何会话变量状态)。



使用 session_unset()函数并不是从存储机制中完全删除会话。如果希望完全删除会话，需要使用函数 session_destroy()。

2) session_destroy()函数

该函数的具体使用语法格式如下所示：

```
bool session_destroy()
```

session_destroy()函数可以从存储机制中完全删除会话，使当前会话失效。但这并不会销毁用户浏览器中的任何 Cookie。如果用户不想在会话之后使用 Cookie，只需要在 php.ini 文件中

将 session.cookie_lifetime 设置为 0。

下面通过一个示例具体演示该函数彻底删除会话的功能，代码如下所示：

```
<?php
    //在请求设置了一个$PHPSESSID 时才试图结束会话
    if(isset($PHPSESSID)){
        $username = "<p/>End of session ($PHPSESSID).";
        session_start();
        session_destroy();
    }else{
        $username = "<p/>There was no session to destroy!";
    }
?>
<html>
    <head>
    <title>彻底销毁会话</title>
   </head>
   <body>
    <?=$username?>
   </body>
</html>
```

8.3.4　实例描述

现在网络发展如此迅速，大家足不出户就可以买到自己需要的各种生活用品，以及完成订外出机票、车票等等。相信大家一定发现了，现在网上购物和在超市购物一样，可以将要买的物品放入购物车，然后到收银台结账付款。这些强大的功能具体是如何实现的呢？本节实例将使用 session 来具体演示购物车的实现。

8.3.5　实例应用

【例 8-3】 使用 session 实现购物车

(1) 新建一个 login.php 文件，在该文件中定义一个用户登录表单，当用户输入用户名、密码之后，可以单击【登录】按钮，登录到商品浏览页面进行购物，代码如下所示：

```
<table width="392" height="249" border="0" cellpadding="0" cellspacing="0">
    <form name="form1" method="post" action="login_success.php">
        <tr>
            <td width="138">用户名：<input name="user" type="text" id="user3"
size="15"></td>
        </tr>
        <tr>
            <td> 密 码 ： <input  name="pass"  type="password"  id="pass2"
size="15"></td>
        </tr>
        <tr>
```

```
            <td><input type="submit" value="登录"/></td>
        </tr>
    </form>
</table>
```

(2) 新建一个 login_success.php 文件，在该文件中获取用户输入的用户名和密码，并将这些用户信息存放到 Session 中，提示用户登录成功。代码如下所示：

```
<?php
    session_start();
    $user=$_POST['user'];                    //获取表单提交过来的用户名
    $pass=$_POST['pass'];                    //获取表单提交过来的密码
    if($user!="" && $pass!=""){              //判断该用户和密码是否正确
        echo "登录成功!";
        $_SESSION['user']=$user;
        $_SESSION['pass']=$pass;
        $_SESSION['producelist']="";         //发给用户一个购物车
        $_SESSION['quatity']="";             //初始化购物车内没有商品
        echo "<meta http-equiv=\"refresh\" content=\"3;url=goods.php\">3 秒钟
转入主页,请稍等......";
    }else{
        echo "<script>alert('登录失败!');history.back();</script>"; ;
    }
?>
```

(3) 新建一个 conn.php 文件，在该文件中连接 MySQL 数据库，代码如下所示：

```
<?php
    $id=mysql_connect('localhost','root','123456');//连接数据库
    mysql_select_db("cart",$id);                   //设置数据库名称
    mysql_query("set names utf8");                 //设置数据库编码
?>
```

(4) 新建一个 goods.php 文件，在该文件中首先使用 include()函数导入连接数据库文件 conn.php，然后判断 session 会话中的 user 和 pass 是否为空，为空提示未登录，并在 3 秒钟后自动跳转到用户登录页面，不为空，将为用户展现商品信息，用户可以进行购物，代码如下所示：

```
<?php
    session_start();
    include("conn.php");
    if($_SESSION['user']=="" && $_SESSION['pass']==""){
        echo "对不起,您没有正确登录本站!!";
        echo "<meta http-equiv=\"refresh\" content=\"3 url=login.php\">3 秒后
转入登录页面......";
    }else{
?>
<table    width="480"    border="1"    cellpadding="0"    cellspacing="0"
background="images/139sss.jpg">
    <form name="form1" method="post" action="">
```

```
        <tr align="center">
            <td  height="25"  colspan="4"><span  class="style1"> 商 品 展 示
</span></td>
        </tr>
    <?php
        $query="select * from goods id order by id desc";
        $result=mysql_query($query);
        if($result==true){
            while($myrow=mysql_fetch_array($result)){
    ?>
        <tr>
            <td width="120" rowspan="2" align="right">
                <img    src="<?php    echo    $myrow[image];?>"    width="115"
height="100">
            </td>
            <td  width="180"  height="50"  align="center"  valign="middle"
class="style2">
                <span class="style2">商品编号: </span>
                <input name="number" type="text" id="number"
                 value="<?php echo $myrow[number];?>" size="12">
            </td>
            <td colspan="2" align="center" valign="middle" class="style2">商
品名称:
                <input name="name" type="text" id="name"
                value="<?php echo $myrow[name];?>" size="12">
            </td>
        </tr>
        <tr>
            <td height="50" align="center" valign="middle" class="style2">
                <span class="style2">商品价格: </span>
                <input name="counts" type="text" id="counts"
                 value="<?php echo $myrow[price];?>" size="10">
            </td>
            <td colspan="2"> </td>
        </tr>
        <tr>
            <td> </td>
            <td height="22" colspan="3" align="center" class="style2">
                <a href="shop_success.php?lmbs=<?php echo $myrow[id];?>">添加
到购物车</a>
            </td>
        </tr>
    </form>
    <?php }
    }?>
</table>
<?php
}?>
```

(5)　新建一个 shop_success.php 文件，在该文件中判断用户是否将同种商品重复添加到购物车中，重复添加商品会提示用户该商品已在购物车中。并跳转到购物车页面，代码如下所示：

```
<?php
    session_start();
    if($_SESSION['user']==""&&$_SESSION['pass'])
    {
        echo "<script>alert('请先登录后购物!');history.back();</script>";
        exit;
    }
    $lmbs=strval($_GET['lmbs']);                    //获取变量的字符串值
    $array=explode("@",$_SESSION['producelist']);  //切开后的字符串返回到数组变量中
    for($i=0;$i<count($array)-1;$i++)
    {
        if($array[$i]==$lmbs)
        {
            echo "<script>alert('该商品已经在您的购物车中!');history.back();</script>";
            exit;
        }
    }
    $_SESSION['producelist']=$_SESSION['producelist'].$_GET['lmbs']."@";
    $_SESSION['quatity']=$_SESSION['quatity']."1@";
    header("location:cart.php");
?>
```

(6)　新建一个 cart.php 文件，显示用户添加到购物车中的商品信息，代码如下所示：

```
<?php
    session_start();
    include("conn.php");
    if($_SESSION['user']==""){
        echo "<script>alert('请先登录,后购物!');history.back();</script>";
        exit;
    }
?>
<table width="480" border="0" align="center" cellpadding="0" cellspacing="0">
    <form name="form1" method="post" action="cart.php">
        <tr>
        <td height="32" background="images/143ss..gif">   
   
            <?php echo $_SESSION['user'];?>
        </td>
        </tr>
        <tr>
            <td bgcolor="#555555"><table width="480" border="0" align="center"
            cellpadding="0" cellspacing="1" background="images/143s..gif">
            <?php
                session_register("total");
                if($_GET['qk']=="yes"){
                $_SESSION['producelist']="";
```

```
                $_SESSION['quatity']=""; }
                $arraygwc=explode("@",$_SESSION['producelist']);
                $s=0;
                for($i=0;$i<count($arraygwc);$i++){
                    $s+=intval($arraygwc[$i]);//获取变量的整数值
                }
                if($s==0){
                    echo "<tr>";
                    echo" <td height='25' colspan='6' bgcolor='#FFFFFF' align=
'center'>
                        您的购物车为空!</td>";
                    echo"</tr>";
                }else{
            ?>
        <tr>
            <td width="120" align="center" bgcolor="#FFFFFF">
                <span class="style1">商品编号</span>
            </td>
            <td width="120" height="25" bgcolor="#FFFFFF" class="style1">
                <div align="center">商品名称</div>
            </td>
            <td width="60" bgcolor="#FFFFFF" class="style1">
                <div align="center">数量</div>
            </td>
            <td width="60" bgcolor="#FFFFFF" class="style1">
                <div align="center">市场价</div>
            </td>
            <td width="60" bgcolor="#FFFFFF" class="style1">
                <div align="center">小计</div>
            </td>
            <td width="60" bgcolor="#FFFFFF" class="style1">
                <div align="center">操作</div>
            </td>
        </tr>
        <?php
            $total=0;
            $array=explode("@",$_SESSION['producelist']);
            $arrayquatity=explode("@",$_SESSION['quatity']);
            while(list($name,$value)=each($_POST)){
            //list 把数组中的值赋给一些变量 ,each 返回数组中
            //当前的键/值对并将数组指针向前移动一步
            for($i=0;$i<count($array)-1;$i++){
                if(($array[$i])==$name){
                    $arrayquatity[$i]=$value;
                }
            }
        }
        $_SESSION['quatity']=implode("@",$arrayquatity);
        for($i=0;$i<count($array)-1;$i++){
```

```
        $lmbs=$array[$i];
        $num=$arrayquatity[$i];
        if($lmbs!=""){
            $sql=mysql_query("select * from goods where id='".$lmbs."'",$id);
            $info=mysql_fetch_array($sql);
            $total1=$num*$info[price];
            $total+=$total1;
            $_SESSION["total"]=$total;
    ?>
    <tr>
        <td height="25" bgcolor="#FFFFFF">
            <div align="center" class="style2">
                <?php echo $info[number];?>
            </div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
            <div align="center" class="style2">
                <?php echo $info[name];?>
            </div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
            <div align="center" class="style2">
                <input type="text" name="<?php echo $info[id];?>"
                    size="2" class="inputcss" value=<?php echo $num;?>>
            </div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
            <div align="center" class="style2">
            <?php echo $info[price];?>元</div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
            <div align="center" class="style2">
            <?php echo $info[price]*$num."元";?></div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
            <div align="center" class="style2">
                <a href="delete.php?id=<?php echo $info[id]?>">移除</a>
            </div>
        </td>
    </tr>
    <?php
        }
    }?>
    <tr>
        <td height="25" colspan="7" bgcolor="#FFFFFF">
            <div align="right">
                <table width="500" height="25" border="0" align="center"
                cellpadding="0" cellspacing="0">
                    <tr class="style1">
```

```
                            <td width="125">
                                <div align="center">
                            <input  type="submit"  value=" 更 改 商 品 数 量 "
class="buttoncss">
                                </div>
                            </td>
                            <td width="125"> </td>
                            <td width="125">
                                <div align="center">
                                    <a href="cart.php?qk=yes">清空购物车</a>
                                </div>
                            </td>
                            <td width="125">
                                <div align="left">总计：<?php echo $total;?></div>
                            </td>
                        </tr>
                    </table>
                </div></td>
            </tr>
            <?php
                }
            ?>
            </table>
        </td>
        </tr>
    </form>
</table>
```

(7) 新建一个 delete.php 文件，在该文件中可以删除用户购物车中的商品，代码如下所示：

```
<?php
    $id=$_GET['id'];
    session_start();
    $arraysp=explode("@",$_SESSION['producelist']);
    $arraysl=explode("@",$_SESSION['quatity']);
    for($i=0;$i<count($arraysp);$i++)
    {
        if($arraysp[$i]==$id)
        {
            $arraysp[$i]="";
            $arraysl[$i]="";
        }
    }
    $_SESSION['producelist']=implode("@",$arraysp);
    $_SESSION['quatity']=implode("@",$arraysl);
    header("location:cart.php");
?>
```

8.3.6　运行结果

打开 IE 浏览器在地址栏中输入“http://localhost/PHP_10/cart/login.php”，进入登录页面，用户输入用户名与密码，即可登录，执行效果如图 8-6 所示。

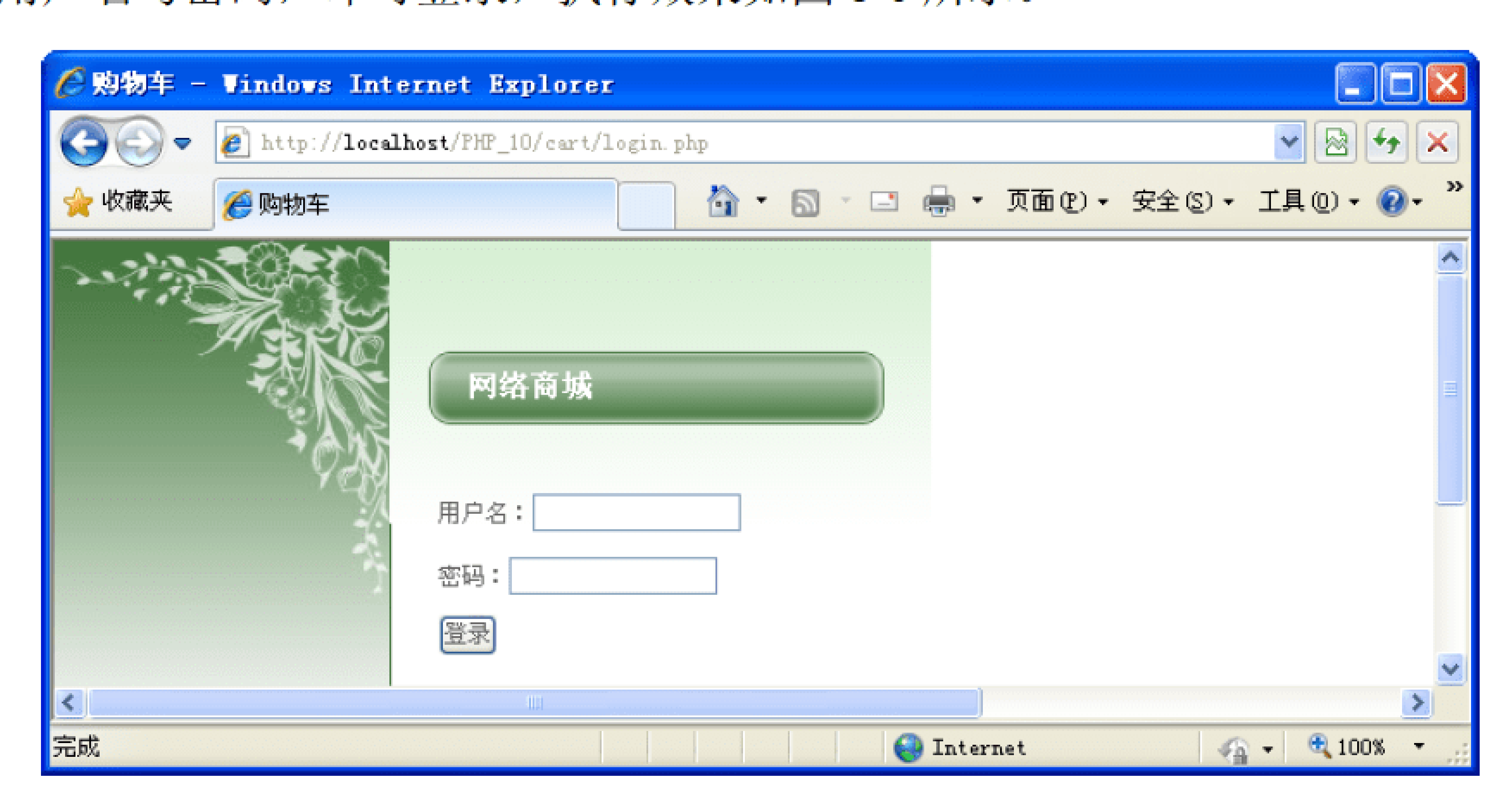


图 8-6　用户登录

登录成功之后，可以进入商品浏览页面，执行效果如图 8-7 所示。



图 8-7　商品信息页面

可以单击【添加到购物车】链接，将商品添加到购物车，执行效果如图 8-8 所示。



图 8-8　购物车信息

也可以单击【清空购物车】链接，将购物车清空，执行效果如图 8-9 所示。



图 8-9　清空购物车

8.3.7　实例分析



源码解析：

本实例首先创建了一个用户登录页面 login.php，当用户输入用户名和密码，登录成功之后，可以看到商品的信息，并单击【添加到购物车】链接，将自己喜欢的商品添加到购物车中，添加完成之后，页面自动跳转到显示购物车信息页面 cart.php，看到自己添加到购物车中的商品，同时可以更改购买商品的数量或将某个商品从购物车中移除。最后清空购物车。

8.4　用 户 登 录

当用户访问某个网站会被分配一个唯一的标识符，即所谓的会话 ID。它要么存放在客户端的 Cookie，要么由 URL 传递。会话允许用户注册任意数目的变量并保留给各个请求使用。

当来客访问网站时，PHP 会自动(如果 session.auto_start 被设为 1)或在用户请求时检查请求中是否发送了特定的会话 ID。本节将具体介绍如何获取与设置会话 ID、会话 ID 的传输、会话存储数据等等。

视频教学：光盘\videos\08\会话基本用法.avi　　长度：11 分钟
光盘\videos\08\会话数据.avi　　长度：3 分钟

8.4.1　基础知识——获取与设置会话 ID

当在 PHP 脚本中第一次调用 session_start()时，会产生一个会话 ID(SID)，这个 ID 会将所有的会话数据绑定到某个特定用户。使用 PHP 可以自动创建和传播会话 ID，但并不是任何情况下都希望使用 PHP 自动创建和传播 ID，有时也需要手工获取和设置会话 ID，使用 session_id()函数能够完成这两项任务。该函数的语法格式如下所示：

```
string session_id([string id]);
```

如果不为 session_id()函数指定参数，则返回当前会话 ID，如果包括可选参数 id，当前会话 ID 将被该值替换。通过下面实例具体讲解 session_id()函数，代码如下所示：

```
<?php
    session_start();                          //自动创建 SID
    echo "PHP 自动产生的 SID：".session_id(); //获取自动创建的 SID
    echo "<hr/>";
    session_id('username');                   //手工设置 SID
    echo "手工编写的 SID：".session_id();     //获取手工设置的 SID
?>
```

执行上述代码，将在页面中输出如下结果：

```
PHP 自动产生的 SID：ns7jrm7g47j6im4o890b384c33
```

```
--------------------------------------------------------------------------------
手工编写的 SID：username
```

8.4.2　基础知识——会话 ID 传输

在 PHP 脚本中第一次调用 session_start()时，会产生一个会话 ID，默认情况下会在响应中包含一个首部字段 Set-Cookie。响应通过名称 PHPSESSION 和会话 ID 的值在浏览器中设置一个会话 Cookie。PHP 会话管理自动地包含该 Cookie，而不需要待用 setCookie()或 header()函数。会话 ID 由 32 个十六进制数字的随机字符串组成。

每当开始一个新会话，PHP 都会创建一个会话文件。默认情况下，会话文件使用会话 ID 加前缀 sess_作为文件名写到/tmp 目录中。会话 ID 的传输有以下两种方式：

- Cookie：用户访问网站，可以获取会话 ID，使用与会话 ID 相关的各种数据。这样一来，即使会话结束，Cookie 也可以在客户端保存，同时也可以在后续会话中读取。

● URL：通过在每个请求页面的本地 URL 上增加会话 ID 可以实现其传输。这样无论用户何时单击这些本地链接，都会使会话 ID 自动传输，这被称为 URL 重写。通过这种方式，即使客户端禁用 Cookie，也不会影响网站会话处理功能的正常使用。

8.4.3 基础知识——使用会话存储数据

创建会话变量可以存储一些数据，会话变量的类型决定了会话要存储的数据。会话变量可以是布尔型、整型、双精度型、字符串型、对象型，或者是这些变量类型的数组。



当使用对象会话变量时必须十分小心，因为初始化一个已有会话时，PHP 需要访问已注册对象的类定义。如果将对象保存为会话变量，用户应该将对象的类定义保存在初始化会话的所有脚本中，而不管脚本是否使用类。

PHP 可以将值串行化(serialize)后，再将会话变量保存在会话文件中。变量的串行化表示将名称、类型和值保存为一个适合于写到文件的字符流，如下所示：

```
username|u:5;register_time|u:123456
```

下面定义两个变量， $username 和$register_time。当会话注册一个会话变量，就设置为从库函数 time()返回的当前时间。还判断会话变量$username 是否已注册，如果已注册直接为会话变量$username 赋值，如果没有注册，则进行注册。还通过计算得出了$username 会话变量持续了多长时间，具体代码如下所示：

```
<?php
    session_start();
    if(!session_is_registered("username"))
    {
        session_register("username");
        $_SESSION['register_time'] = time();
        $username="";
    }else{
        $username = "itzcn";
    }
    $sessionId = session_id();
    echo "会话 Id: ".$sessionId;
    echo "<br/>count=".$count;
    echo "<br/>session 变量开始时间=".$_SESSION['register_time'];
    $continue_time = time() - $_SESSION['register_time'];
    echo "<br/>该 session 值持续了".$dur."秒";
?>
```

8.4.4 基础知识——编码和解码会话数据

PHP 是以标准化格式来存储数据的，需要注意的是各个会话变量之间都以分号隔开，每个会话都由 3 部分组成：名、长度和值。存储会话数据的基本语法格式如下所示。

```
name | s:length:"value"
```

下面来看两个变量，这两个变量在 PHP 会话中存储，代码如下所示：

```
username | 5:"admin";password | 6:"123456";
```

PHP 提供了自动处理会话的编码和解码，但同获取和设置会话 ID 一样，在有些情况下不希望自动处理会话的编码和解码，而希望手工处理会话的编码和解码，这可以通过 session_encod()和 session_decode()函数来实现。

1．session_encode()编码函数

该函数可以手工将所有会话变量编码为一个字符串，然后将此字符串插入到数据库，并在获取之后使用 session_decode()对其进行解码。session_encode()函数语法格式如下所示：

```
string session_encode()
```



session_encode()函数将对该用户的所有会话变量编码，而不只是对执行 session_encode()的脚本中注册的会话变量进行编码。

由 session_encode()函数的语法格式可知，该函数的返回值为编码后的字符串。下面通过一个实例创建一个会话变量并赋值，然后使用 session_encode()函数编码，并获取编码后的字符串，代码如下所示：

```
<?php
    session_start();
    $_SESSION['username'] = "xiaoxiao";
    $_SESSION['password'] = "123456";
    $userinfo = session_encode();
    echo $userinfo;
?>
```

执行上述代码，将在页面中输出如下结果：

```
username|s:8:"xiaoxiao";password|s:6:"123456";
```

2．session_decode()解码函数

会话数据通过编码会话函数编码之后，返回的字符串可以通过 session_decode()函数进行解码，解码之后得到变量最初的格式，如果解码执行成功则返回 true，否则返回 false。session_decode()函数的语法格式如下所示：

```
bool session_decode(string $userstring)
```

下面创建两个会话变量 username 和 truename，使用 session_encode()函数对会话变量数据进行编码，然后再使用 session_decode()函数进行解码，代码如下所示：

```
<head>
<!-- 设置编码避免出现乱码 -->
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
```

```
<?php
    session_start();
    $_SESSION['username'] = "xiaoxiao";
    $_SESSION['truename'] = "晓晓";
    $userinfo = session_encode();           //编码会话变量
    echo "编码之后用户数据：".$userinfo;      //输出编码后的数据
    echo "<br/>";
    echo "用户名：".$_SESSION['username'];   //输出用户名
    echo "<br/>";
    echo "真实姓名：".$_SESSION['truename']; //输出真实姓名
    echo "<br/>";
    session_decode($userinfo);              //解码会话变量
    echo "解码后变量数据：<br/>";
    echo "用户名：".$_SESSION['username'];   //输出用户名
    echo "<br/>";
    echo "真实姓名：".$_SESSION['truename']; //输出真实姓名
?>
```

执行上述代码，在页面中输出如下结果：

```
编码之后用户数据：username|s:8:"xiaoxiao";password|s:6:"123456";truename|s:6:"
晓晓";
用户名：xiaoxiao
真实姓名：晓晓
解码后变量数据：
用户名：xiaoxiao
真实姓名：晓晓
```

8.4.5 实例描述

网站一般在用户第一次登录时，会将用户的一些信息存储到 session 中，以跟踪用户登录后执行的一些其他操作，但具体如何在用户第一次登录时将用户信息存储在 session 中呢？本节实例将具体讲解如何在用户登录时存储用户信息。

8.4.6 实例应用

【例 8-4】 用户登录

(1) 新建一个 index.php 页面，在该页面中定义一个用户登录表单，当用户输入用户名与密码，单击【登录】按钮，即可进行登录，代码如下所示：

```
<form method=post action="login.php">
    用户名：<input type="text" name="name"/><br/><br/>
    密码：<input type="password" name="pwd"/><br/>
    <input type="submit" value="登录">
</form>
```

(2) 新建一个 login.php 页面，在该页面中获取用户输入的用户名与密码，判断输入数据是否合法，并将用户名与密码注册到 session 中，代码如下所示：

```
<?php
    global $login;
    if ($_POST['name']=="" || $_POST['pwd']=="")
    {
        echo "登录失败";
        echo "请<a href=index.html>返回</a>";
        exit;
    }
    $login = array('name'=>$_POST['name'],
            'pwd'=>$_POST['pwd']);
    session_start();
    session_register("login");
    if (session_is_registered('login'))
    {
        echo "hello,".$login['name']."<br/>";
        echo "<a href=logout.php>退出登录</a><br/>";
    }
    else
    {
        echo "非法操作<br/>";
        exit;
    }

?>
```

(3) 当用户输入的单击 login.php 页面中的【退出登录】链接时，进入 logout.php 页面，将注册到 Session 中的用户名与密码销毁掉。代码如下所示：

```
<?php
    session_start();
    session_unset (login);
    header("location:index.html");
?>
```

8.4.7　运行结果

打开 IE 浏览器在地址栏中输入“http://localhost/PHP_10/session/index.php”，进入登录页面，执行效果如图 8-10 所示。

用户输入用户名与密码之后，单击【登录】按钮，登录之后显示问候用户信息及【退出登录】链接，执行效果如图 8-11 所示。

当用户单击【退出登录】链接之后，将跳转到用户登录页面。

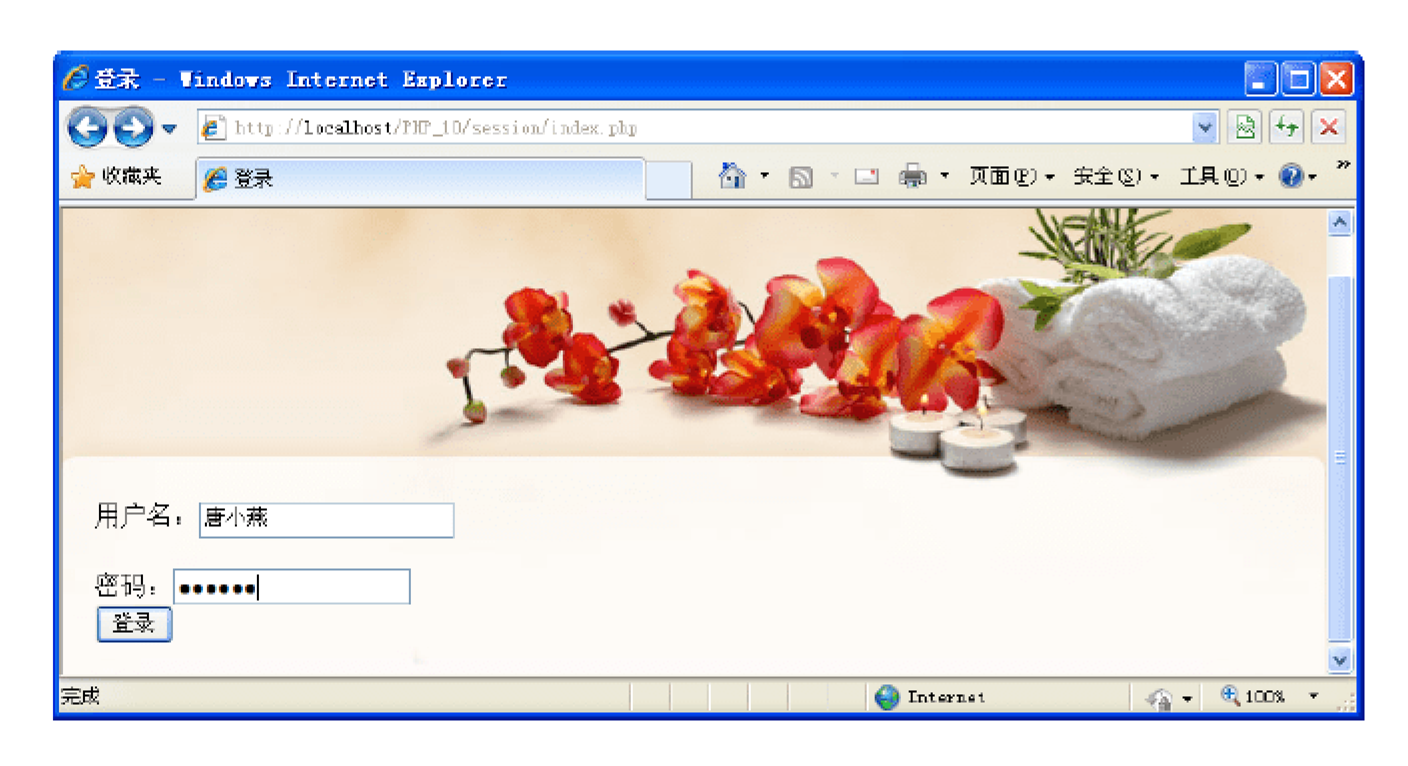


图 8-10　用户登录页面

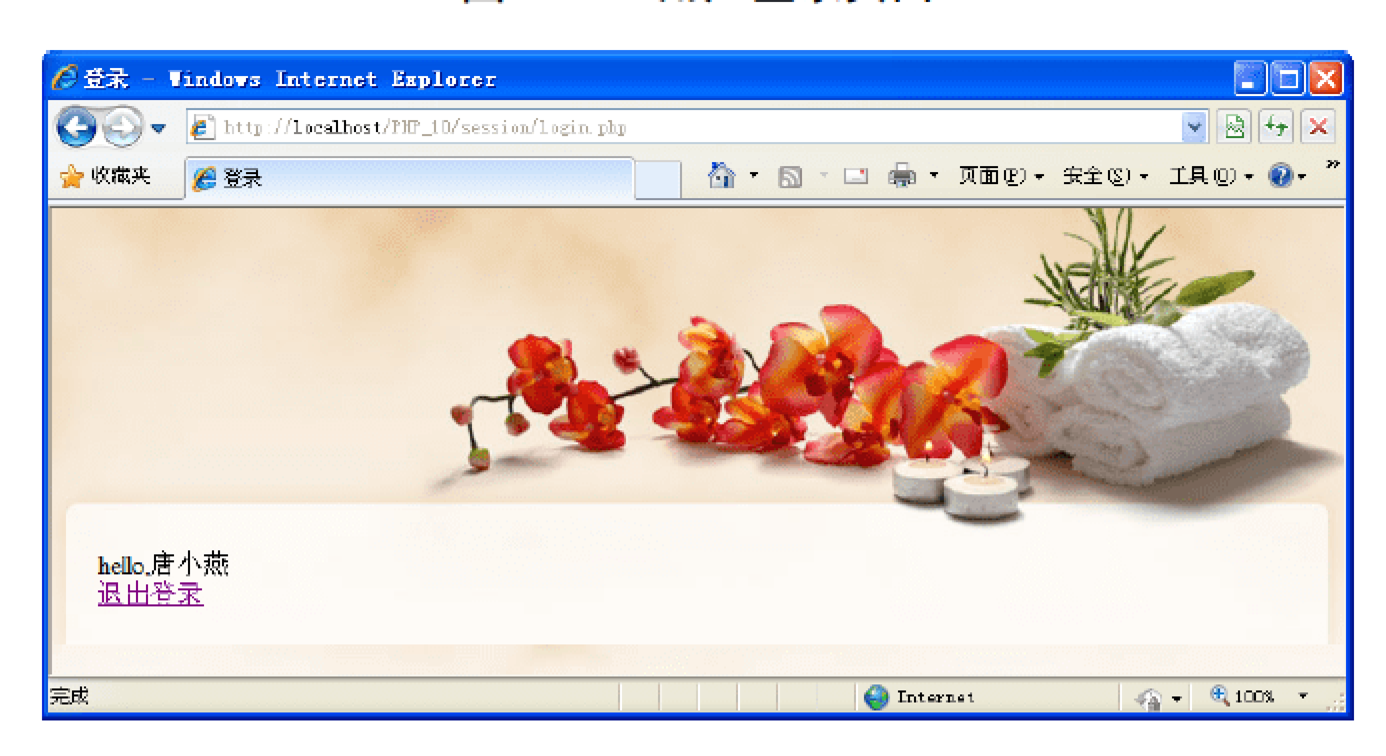
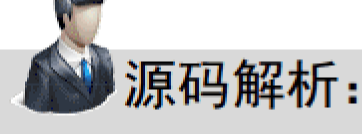


图 8-11　退出登录页面

8.4.8　实例分析



源码解析：

本实例首先创建了一个 index.php 页面用于用户登录，定义了一个用户登录表单，用户可以输入用户名与密码进行登录，并将表单用户信息提交给 login.php 页面处理，将用户名与密码使用 session_register()函数注册到 Session 中，并从 Session 中取出用户名，在页面中显示“hello，用户名”。当用户单击【退出登录】链接时，进入 logout.php 页面，使用 session_unset()函数将用户信息从 Session 中移除掉。

8.5　常见问题解答

8.5.1　PHP Cookie 获取不到的情况



PHP Cookie 获取不到怎么办？

网络课堂：http://bbs.itzcn.com/thread-12426-1-1.html

以下代码：

```
<?php
    setcookie("xiami","cookietest");
    if(empty($_COOKIE["xiami"])){
        echo '<script>alert("您的浏览器没有开启 Cookie,部分功能将访问出错！
");</script>';
        die('显示 cookie 的开启方法');
    }
?>
```

我在判断之前设置了 Cookie，为什么还是获取不到呢？求高手解决。

【解决办法】：Cookie 在当前页面是不能用的，也就是你不能设置了 Cookie 就接着使用。Cookie 的值是存在于客户端的浏览器之中的，所以设置之后，需要第二次发出请求才可以使用这个值。

8.5.2　如何删除 Cookie



如何删除 Cookie?

网络课堂：http://bbs.itzcn.com/thread-12427-1-1.html

我用下面的方式设置 Cookie，代码如下：

```
<?php
    // 设定 cookie
    setcookie("cookie[three]", "cookiethree");
    setcookie("cookie[two]", "cookietwo");
    setcookie("cookie[one]", "cookieone");

    // 刷新页面后,显示出来
    if (isset($_COOKIE['cookie']))
    {
        foreach ($_COOKIE['cookie'] as $name => $value)
    {
            echo "$name : $value <br />\n";
        }
    }
?>
```

我想问怎样删除 Cookie，怎样获取其中 username 的值？

【解决办法】：获取 Cookie 里面 username 的值的方法如下。

```
echo $_COOKIE['username'];
```

删除 Cookie 只需要设置为空值，例如：

```
setcookie("cookie[three]", "");
```

删除所有的 Cookie 语句如下：

```
foreach ($_COOKIE as $key=>$val)
setcookie($key, "");
```

8.5.3 PHP 中 Session 验证的问题



PHP 中 Session 验证的问题。
网络课堂：http://bbs.itzcn.com/thread-12428-1-1.html

我在写一个 Session 验证登录的程序，可我写好后发现要登录的页面提示是无法浏览，找了好久没有找到问题所在，请高手指点一下。

下面是我做的验证用户登录的程序。

```
<?php
    $user=$_POST[user];
    $pw=$_POST[pw];
    echo $pw."<br>";   echo $user;
    $sql="select *  from user where name='$user'";
    $query=mysql_query($sql);
    $result=mysql_fetch_array($query);
    if(isset($result))
    {
        if($result['pssword']==$pw)
        {
            session_start();
            // 注册登录成功的 admin 变量,并赋值 true
            $_session['admin'] = true;
            header("location:admin_index.php");
            exit;
        }
    }
    else
        echo "<script language=javascript>alert('对不起！用户名或密码错误！');
        window.location.href='login.php';</script>";
        exit;
?>
```

下面是我写的检查是否登录的问题。如果登录才能浏览本页，否则是浏览不了的。

```
<?php
    $admin=false;
    session_start();
    include("conn.php");
?>
<?php
    // 判断是否登录
    if (isset($_SESSION['admin']) && $_session['admin'] ===true)
    {
        echo "您已经成功登录";
    }
    else
```

```
    {
    // 验证失败，将 $_session["admin"] 置为 false
        $_session['admin'] = false;
        die("您无权访问");
    }
?>
```

登录时提示“您无权访问”，请指点。

【解决办法】：这个没有什么大问题，就是代码中有一点错误。

```
session_start();
// 注册登陆成功的 admin 变量,并赋值 true
$_session['admin'] = true;
```

你的这个 Session 根本传不过去，要注意大小写啊，要改成：

```
$_SESSION['admin'] = true;
```

8.5.4　PHP Session 写入问题与登录验证



PHP Session 写入问题与登录验证。

网络课堂：http://bbs.itzcn.com/thread-12429-1-1.html

用 PHP 编了一个登录验证的小程序，可运行时总提示如下的错误。

```
Warning: session_register() [function.session-register]: Cannot send session
cookie - headers already sent by (output started at E:\网站程序
\czj\include\conn.inc.php:5) in E:\网站程序\czj\accbm\ii.php on line 48
3 秒钟后转入主页,请稍等......
```

代码如下。

```
    //登录开始
    $user=$_POST[bmzid];
    $pass=$_POST[pws];
    //建立查询 SQL 语句
    $sqlst1 = "select * from bm_form_bm WHERE bmzid = '$user' and prss = '$pass' ";
    $result1 = mysql_query($sqlst1,$conn); //开始查询数据库
    while ($rows = mysql_fetch_array($result1)){
        $bmzid = $rows[bmzid];
        $prss = $rows[prss];
    }

    if($user=="$bmzid" && $pass=="$prss"){ //判断该用户和密码是否正确
        echo "登录成功!";
        session_register("user");           //注册新的 session 变量
        session_register("pass");
        echo "<meta http-equiv='refresh' content='3;url=ii.php?user=a'>3 秒钟
后转入主页,请稍等......";
    }else{
        echo "登录失败!!";
```

```
        echo "<meta http-equiv=\"refresh\" content=\"3;url=login.php\">3 秒钟
后转入前页,请稍等......";
        exit;
    }
//登录结束
```

数据库连接文件 conn.inc.php 是这样写的。

```
<?php
    $conn = mysql_connect("localhost","root","admin");
    mysql_query("set names gb2312");
    mysql_select_db("bm");
?>
```

错在哪里呀？是不是 PHP 配置上有问题呀？

【解决办法】：headers already sent by 表示 session_register 前有输出。

conn.inc.php 应该是最后还有一空白行输出删掉建议连同最后一个分界符一起删掉，即

```
<?php
    $conn = mysql_connect("localhost","root","admin");
    mysql_query("set names gb2312");
    mysql_select_db("bm");
```

同时预防第二个错误。

```
if($user=="$bmzid" && $pass=="$prss"){     //判断该用户和密码是否正确
    session_register("user");              //注册新的 session 变量
    session_register("pass");
    echo "登录成功!";
    echo "<meta http-equiv='refresh' content='3;url=ii.php?user=a'>3 秒钟后转
入主页,请稍等......";
}
```

要先 Session 后输出。

8.6 习　　题

一、填空题

(1) 通过下面代码可以看出，在 Session 启动以后，可以使用 PHP 的__________变量来获取和设置 Session 变量。

```
<?php
session_start();
$_SESSION['views']=1;
?>
```

(2) 在向 PHP Session 中保存信息之前，需要首先使用 Session_start()函数来启动 Session，这个函数必须在<html>标签之前调用。代码划线处应填写__________。

```
<? php
    __________
?>
<html>
    <body>
    </body>
</html>
```

(3) 如果需要删除一些 Session 数据，可以使用 unset()函数把某个 Session 变量的值清空，但是如果需要把整个 Session 销毁应使用__________函数。

二、选择题

(1) 下面代码，创建一个 Cookie，正确的是__________。

A. `setcookies("username","Jack");`

B. `setcookies("password","123456",time()+7200);`

C. `setcookie("cookie","cookie value",time+7200,"/forum",".phpuser.com",1);`

D. `setcookies();`

(2) 删除一个 Cookie，下面代码正确的是__________。

A. `setcookie("password","123456",time()-2);`

B. `setcookie("password","123456");`

C. `setcookie("password","123456",time()+2);`

D. `setcookies("password","123456",time()+2);`

(3) 如何使用 PHP 为会话变量编码，下面代码正确的是__________。

A.
```
<?php
session_start();
$_SESSION['customer'] = "xiaoxiao";
$userinfo = session_encode();
echo $userinfo;
?>
```

B.
```
<?
$_SESSION['customer'] = "xiaoxiao";
$userinfo = session_encode();
echo $userinfo;
?>
```

C.
```
<?php
session_start();
$_SESSION['customer'] =xiaoxiao;
$userinfo = session_encode();
echo $userinfo;
?>
```

D.
```
<?php
```

```
session_start();
$_SESSION[customer] =xiaoxiao;
$userinfo = session_code();
echo $userinfo;
?>
```

三、上机练习

上机练习： 通过 Session 实现计数器并禁止页面刷新。

练习要求：开启 session，并定义一个临时会话变量，当临时会话变量为空，表示第一次登录，为临时会话变量赋值。否则的话，弹出对话框提示用户不可以刷新页面，并将页面窗口关闭。声明一个统计页面访问量的计数器变量，用户每访问一次变量加 1，执行效果如图 8-12 所示。

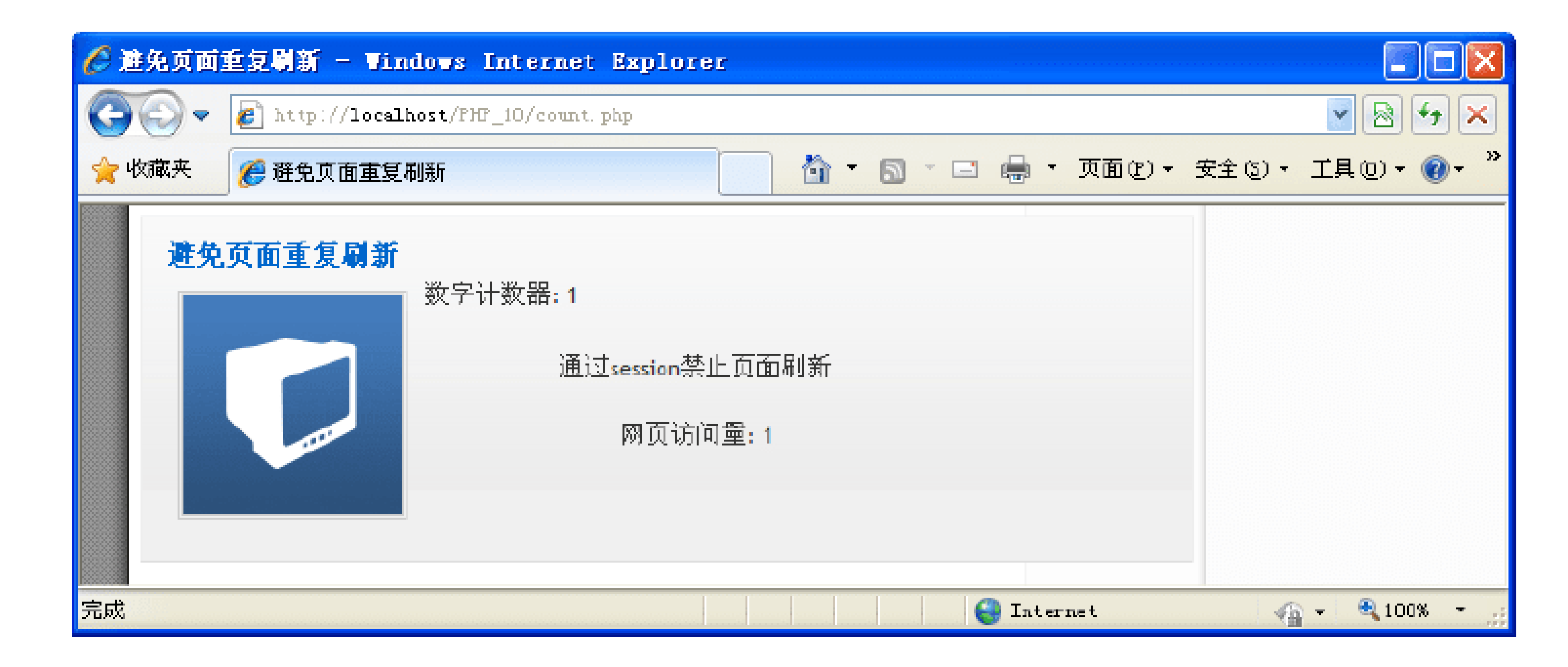
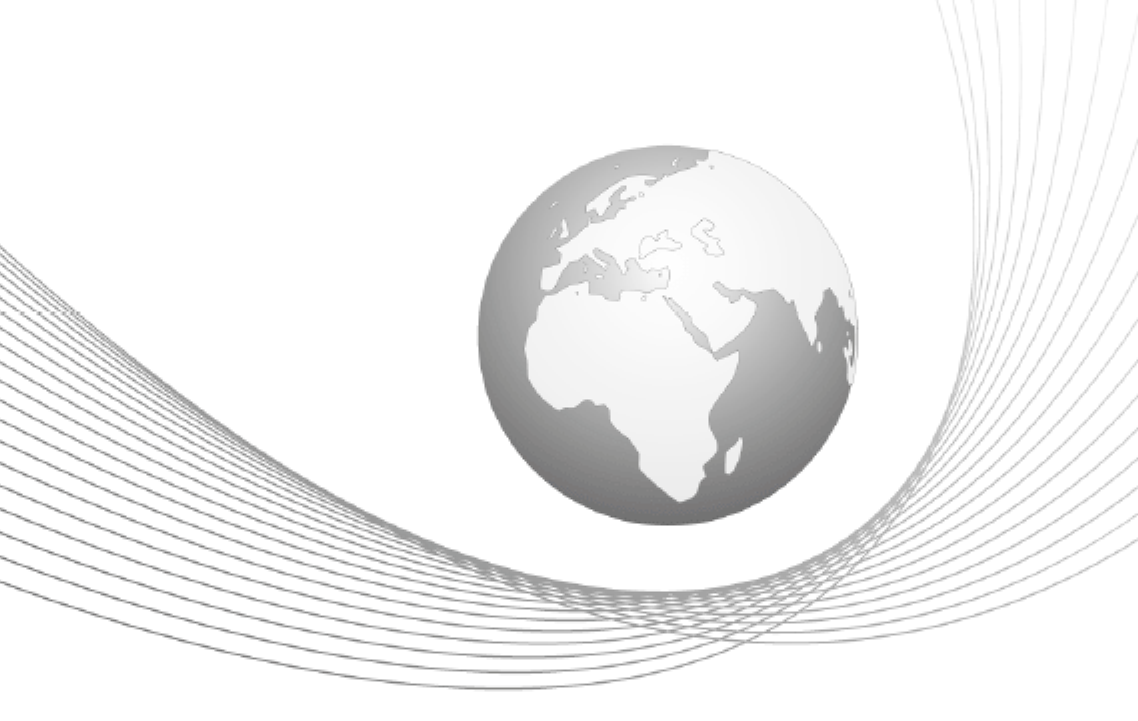


图 8-12　避免页面重复刷新



第 9 章　PHP 操作 MySQL 数据库

内容摘要：

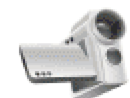
在实际的应用中，通常使用数据库存储数据信息。虽然可以直接对数据库进行操作，但是对于 Web 系统而言，更多的是使用程序对数据库进行相关操作。例如，读取数据库信息、从数据库查询信息等。本章主要介绍 PHP 操作与管理 MySQL 数据库的相关信息。

学习目标：

- 掌握安装 MySQL 的过程。
- 熟练掌握 MySQL 数据库。
- 掌握如何在 PHP 中执行 SQL 语句。
- 掌握获取数据和显示数据的函数。
- 熟练管理数据库中的数据。
- 了解多个查询、准备语句和事务处理。

9.1　安装和配置 MySQL

当今信息社会，各种各样的数据信息占据着重要地位，而数据库能够保存和管理这些数据信息。因此，数据库在现代语言编程中也占有重要的位置，它的安装和配置也随之显得十分重要。本节将讲述 MySQL 数据库的安装和配置。



视频教学：光盘\videos\09\安装 MySQL.avi　　长度：5 分钟

9.1.1　基础知识——安装 MySQL

由于本书是基于 MySQL 数据库介绍 PHP，所以首先将介绍如何安装 MySQL 数据库，其详细步骤如下。

(1) 从 http://dev.mysql.com/downloads/mysql/5.1.html 网址上下载最新版本的 MySQL 安装程序。双击下载好的 mysql-5.5.8-win32.msi 图标，弹出欢迎安装 MySQL 对话框，如图 9-1 所示。

(2) 单击 Next 按钮，弹出是否同意安装协议对话框，启用对话框底部的 I accept the terms in the License Agreement 复选框，选择是否同意安装协议对话框，如图 9-2 所示。



图 9-1　欢迎安装 MySQL 对话框

图 9-2　是否同意安装协议对话框

(3) 单击图 9-2 中的 Next 按钮，弹出选择安装类型对话框，主要有 3 个选择项：标准安装、安全安装和自定义安装。在这里选择自定义安装，选择类型对话框如图 9-3 所示。

(4) 单击选择安装类型对话框中的 Custom 按钮，进入选择组件对话框，如图 9-4 所示。

(5) 在选择组件对话框中选择所有的组件，单击 Browse 按钮，改变安装路径，这里使用的安装路径为“E://MySQL”，然后单击 Next 按钮，进入准备安装对话框，如图 9-5 所示。

(6) 单击准备安装对话框中的 Install 按钮，开始安装 MySQL，如图 9-6 所示。安装过程中，会弹出 MySQL Enterprise 对话框，如图 9-7 所示。

(7) 单击 MySQL Enterprise 对话框中的 Next 按钮，弹出如图 9-8 所示的对话框。

(8) 单击图 9-8 中的 Next 按钮，弹出完成安装对话框，如图 9-9 所示。在单击 Finish 按钮

之前，要确保启用了 Launch the MySQL Instance Configuration Wizard 复选框。

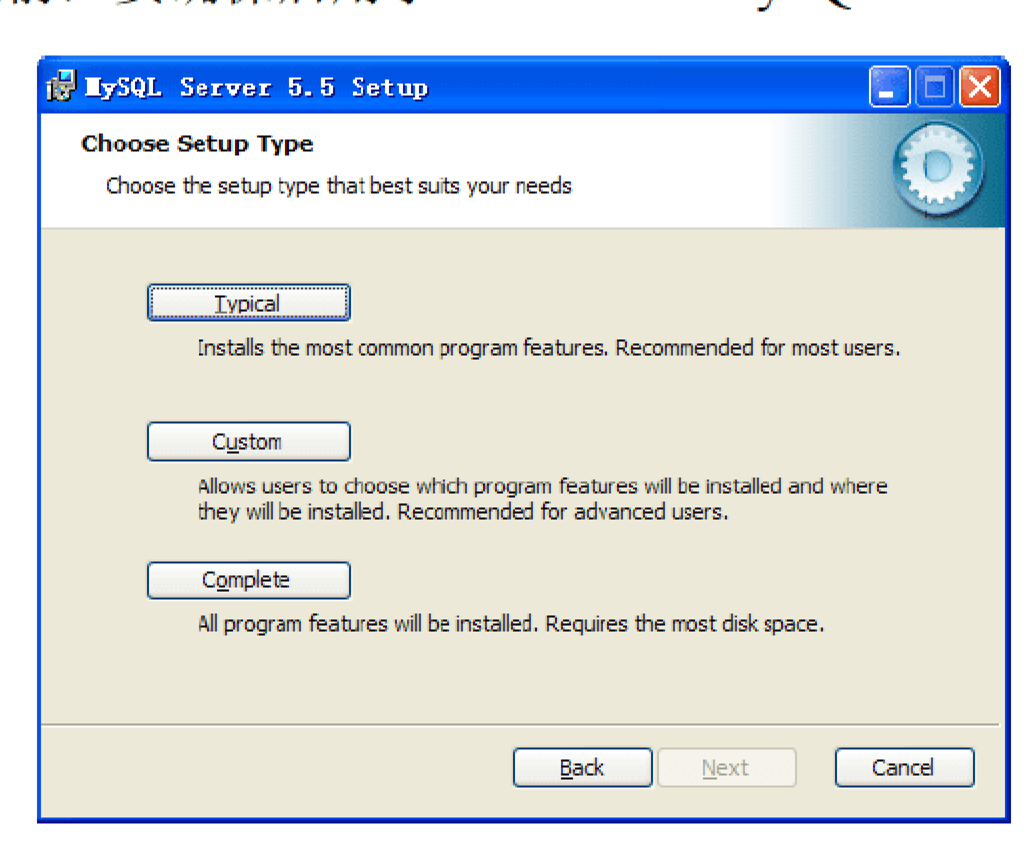


图 9-3　选择安装类型对话框

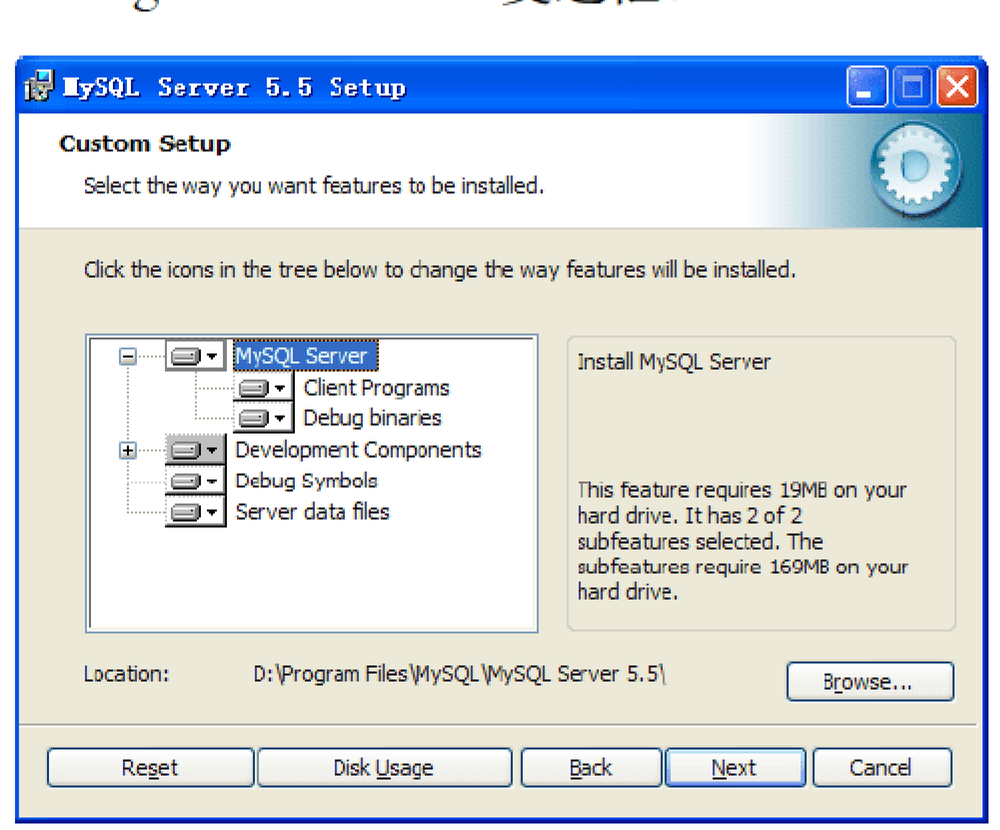


图 9-4　选择组件对话框

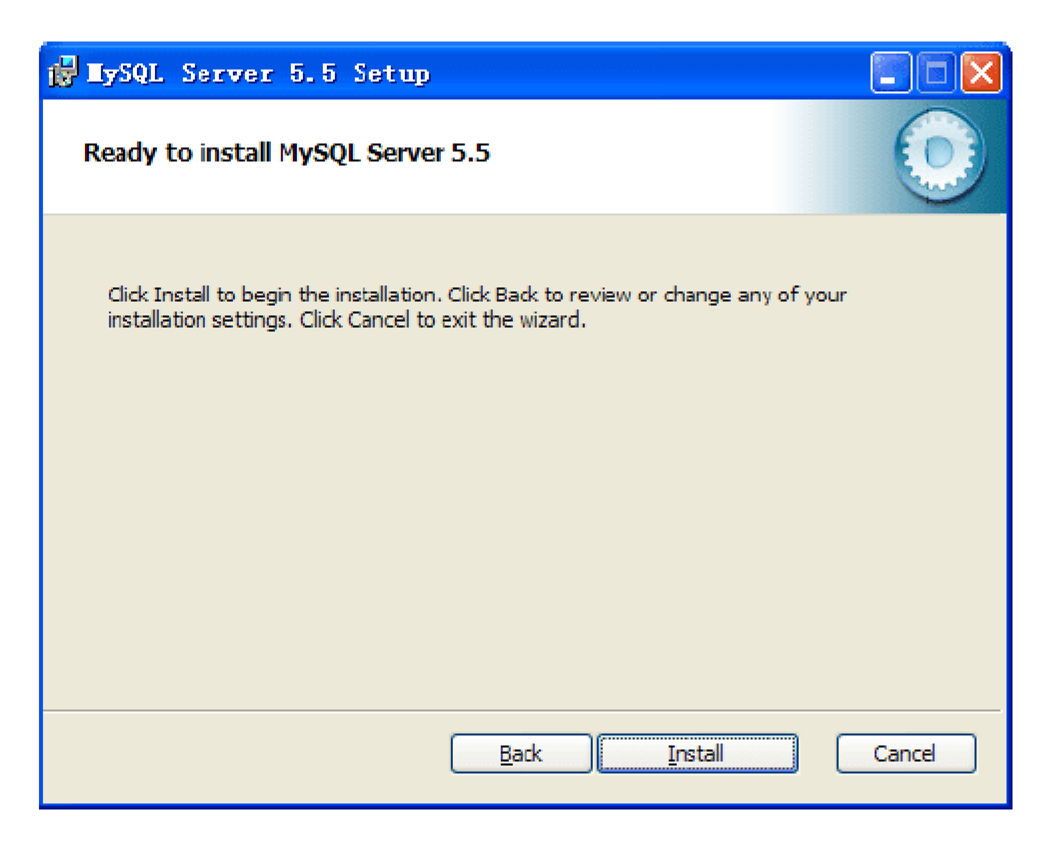


图 9-5　准备安装对话框

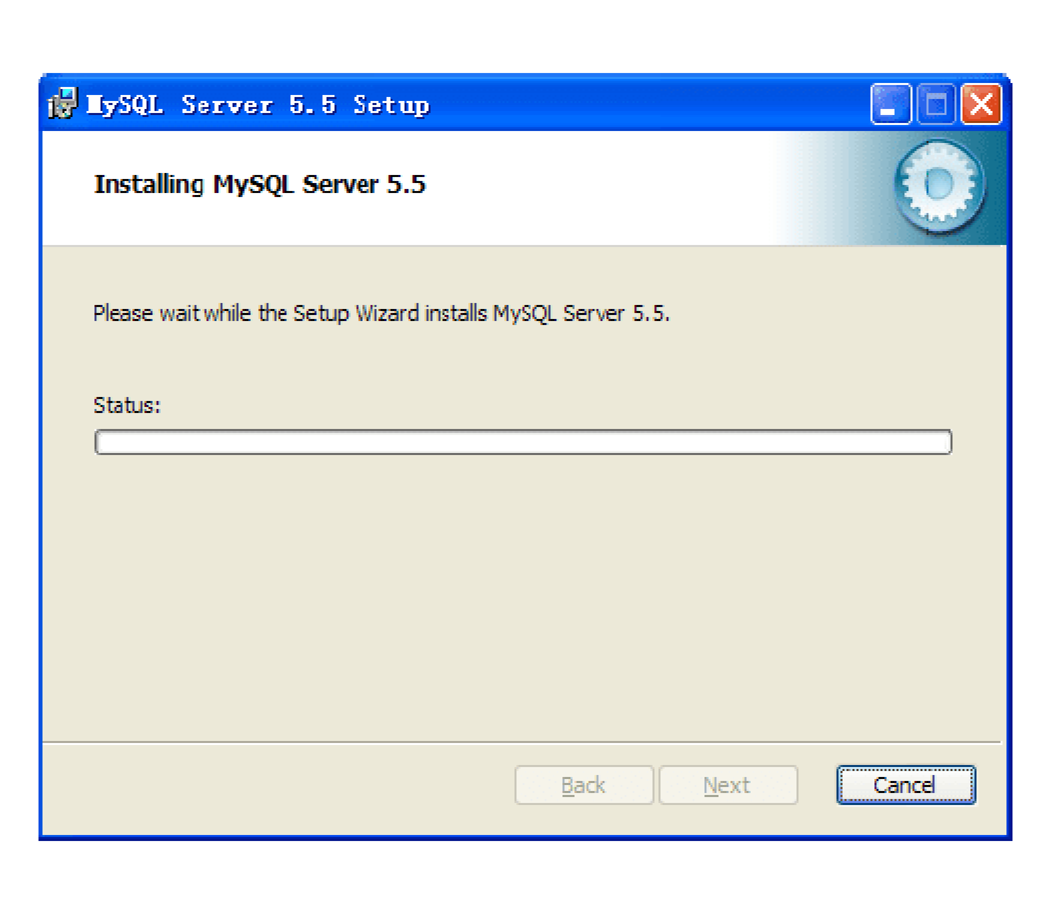


图 9-6　安装 MySQL 对话框



图 9-7　MySQL Enterprise 对话框



图 9-8　MySQL Enterprise Monitor Service 对话框

(9)　在启用 Launch the MySQL Instance Configuration Wizard 复选框的情况下，单击 Finish 按钮，会弹出欢迎配置 MySQL 服务的对话框，如图 9-10 所示。

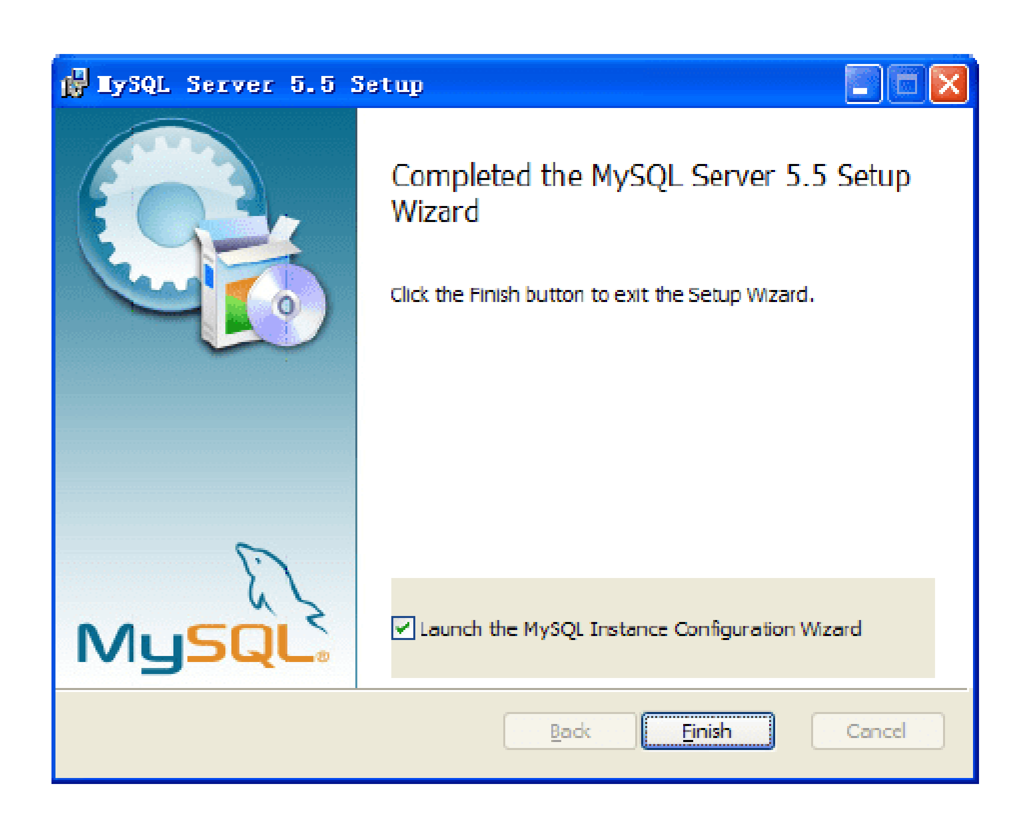


图 9-9　完成安装对话框

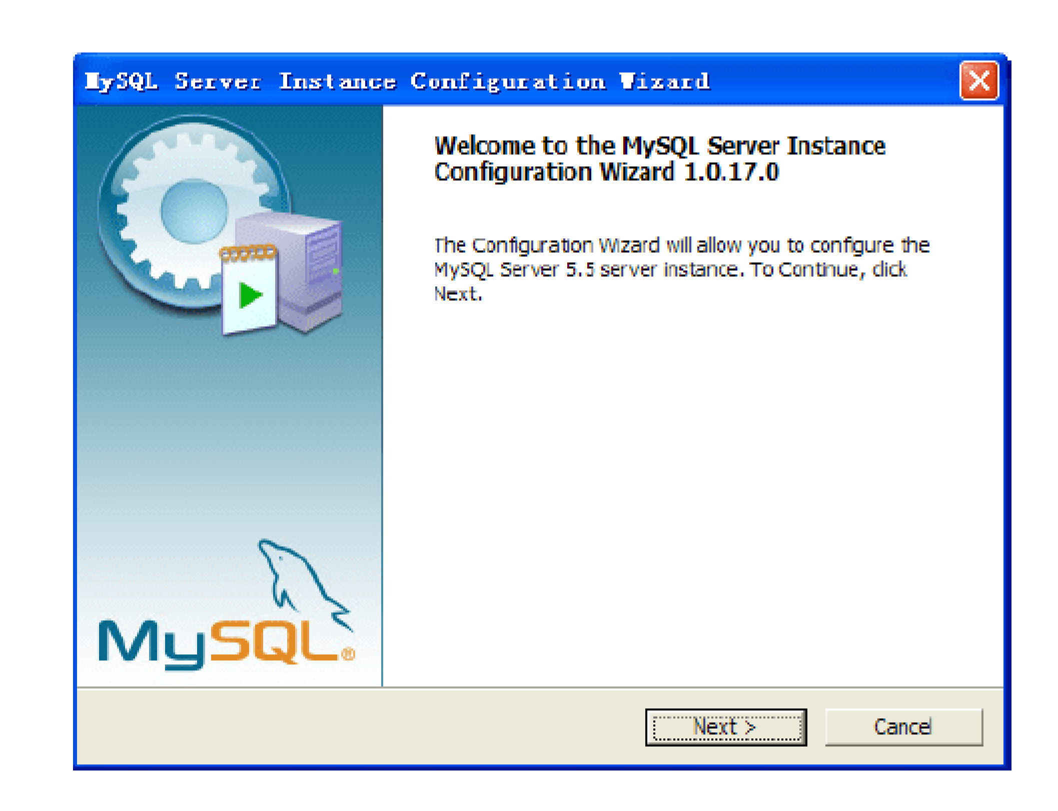


图 9-10　欢迎配置 MySQL 服务对话框

(10) 单击欢迎配置 MySQL 服务对话框中的 Next 按钮，会弹出选择配置类型对话框，如图 9-11 所示。在选择配置类型对话框中有两个选择项：一是表示详细配置；另一个表示标准配置。在这里选择第一个，然后单击 Next 按钮，弹出选择使用服务类型的对话框，如图 9-12 所示。

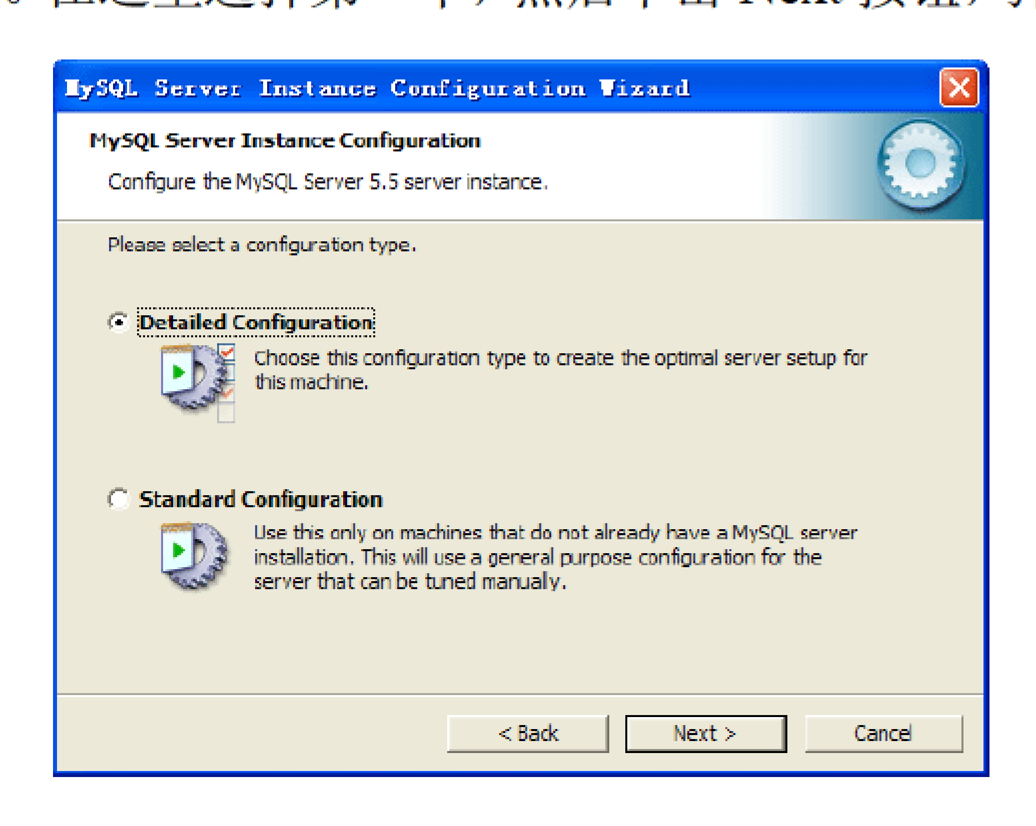


图 9-11　选择配置类型对话框

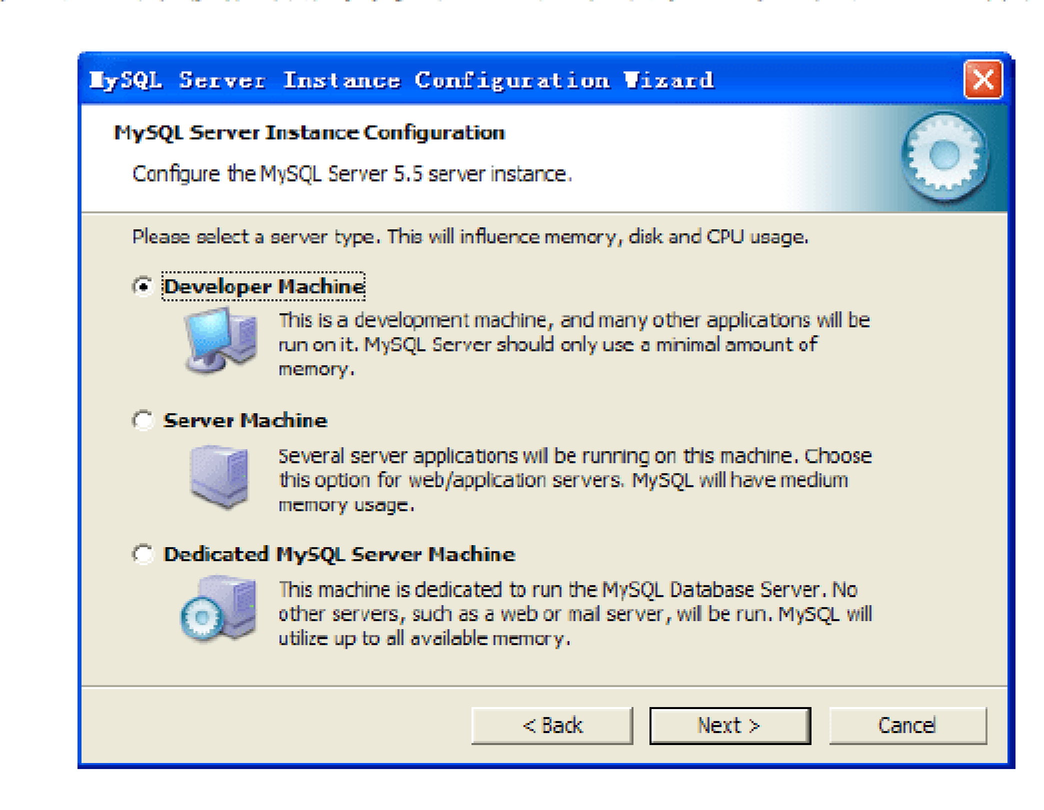


图 9-12　选择服务类型对话框

(11) 单击选择服务类型对话框中的 Next 按钮，会弹出选择使用数据库类型的对话框，如图 9-13 所示，在这里直接使用默认值。然后单击 Next 按钮，进入为数据库选择存储驱动器的对话框，如图 9-14 所示。

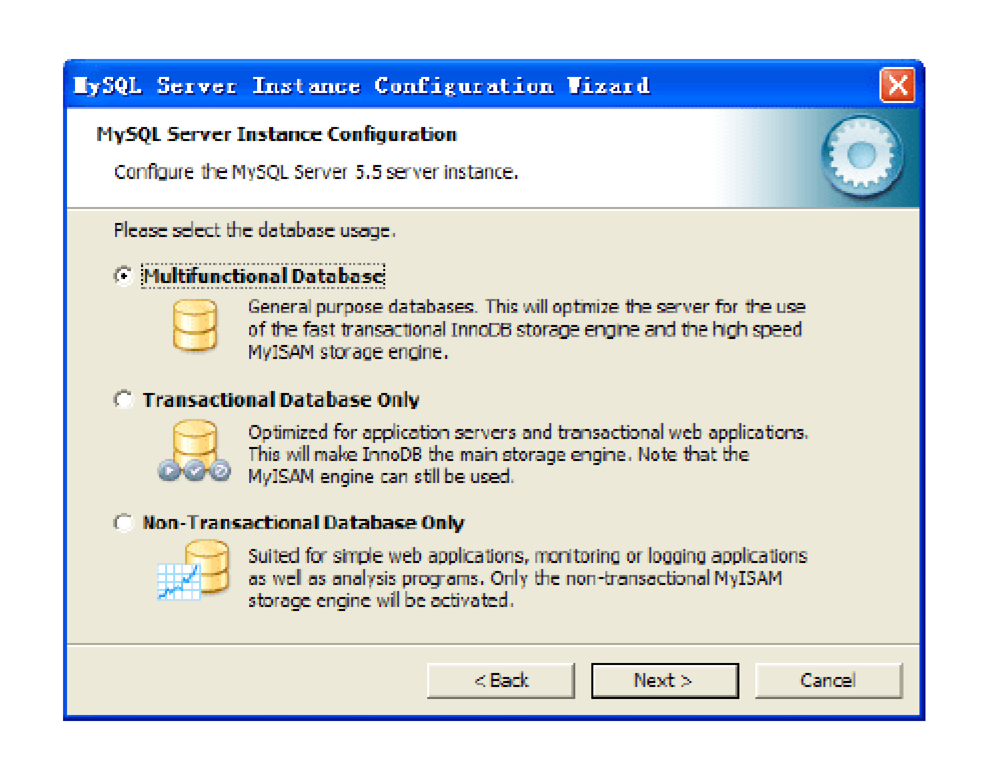


图 9-13　选择所使用的数据库

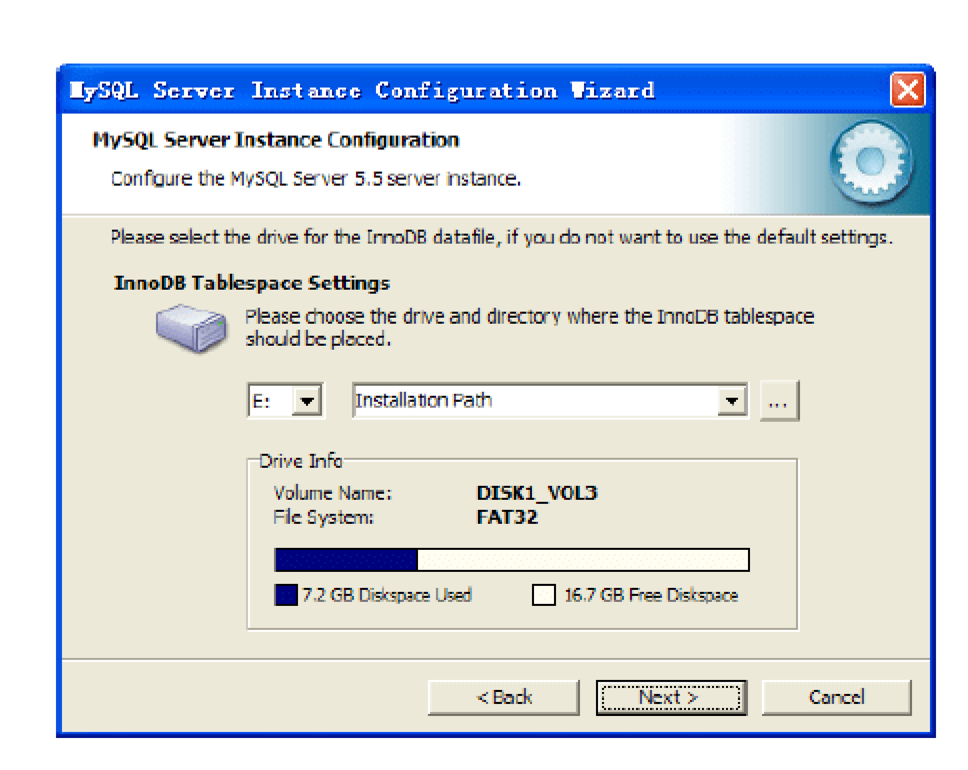


图 9-14　选择存储数据库的驱动器

(12) 选择存储数据库的驱动器后，单击 Next 按钮，弹出选择同时最多执行的线程数对话框，如图 9-15 所示。这里同样选择默认值，即 Decision Support(DSS)/OLAP 选项。然后单击 Next 按钮，弹出选择数据库端口的对话框，如图 9-16 所示。这里使用 3307 端口号，默认为 3306。

(13) 单击选择数据库端口号对话框中的 Next 按钮，进入选择字符集对话框，如图 9-17 所示。在这里选择最后一个选项，并且选择 utf8 字符集。



这里一定要选择 uft8 字符集，否则创建 PHP 应用程序时无法显示数据库中的汉字。

(14) 选择字符集后，单击 Next 按钮，弹出配置 Windows 选项对话框，如图 9-18 所示。在这里选中 Include Bin Directory in Windows PATH 选项，该选项表示将 MySQL 路径添加到 Windows 的环境变量中。

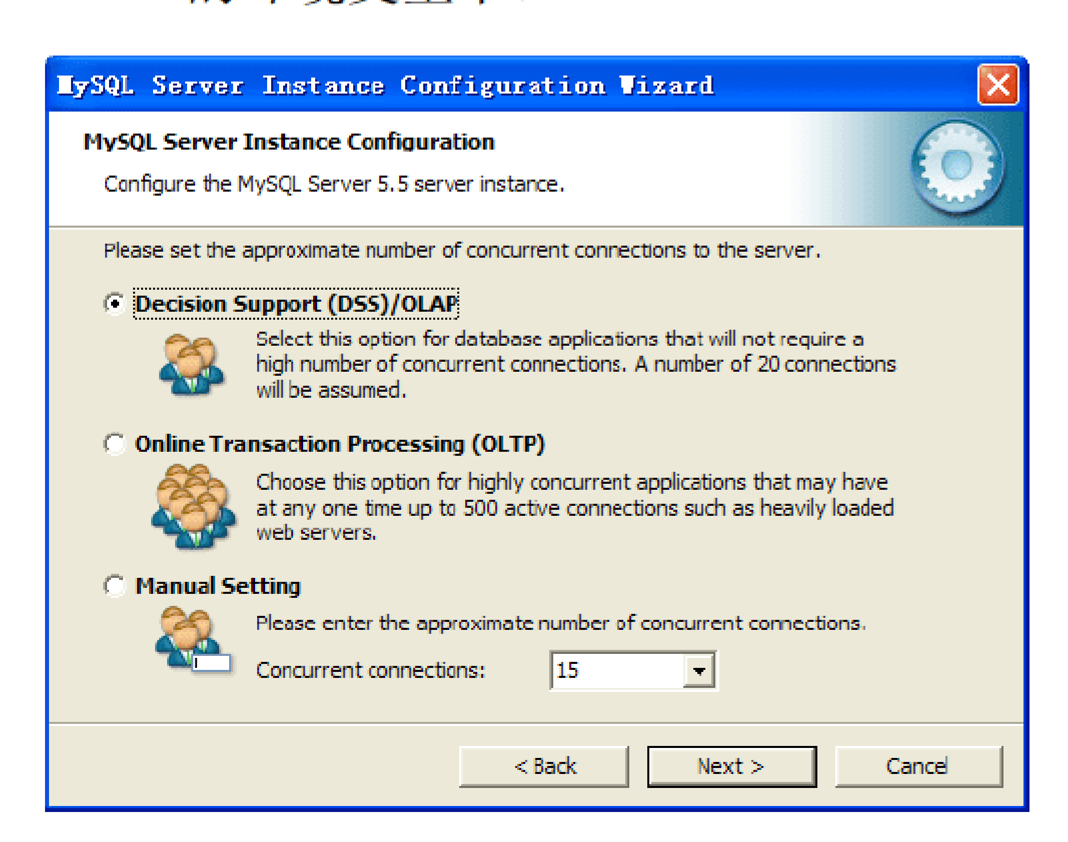


图 9-15　选择同时最多执行的线程数

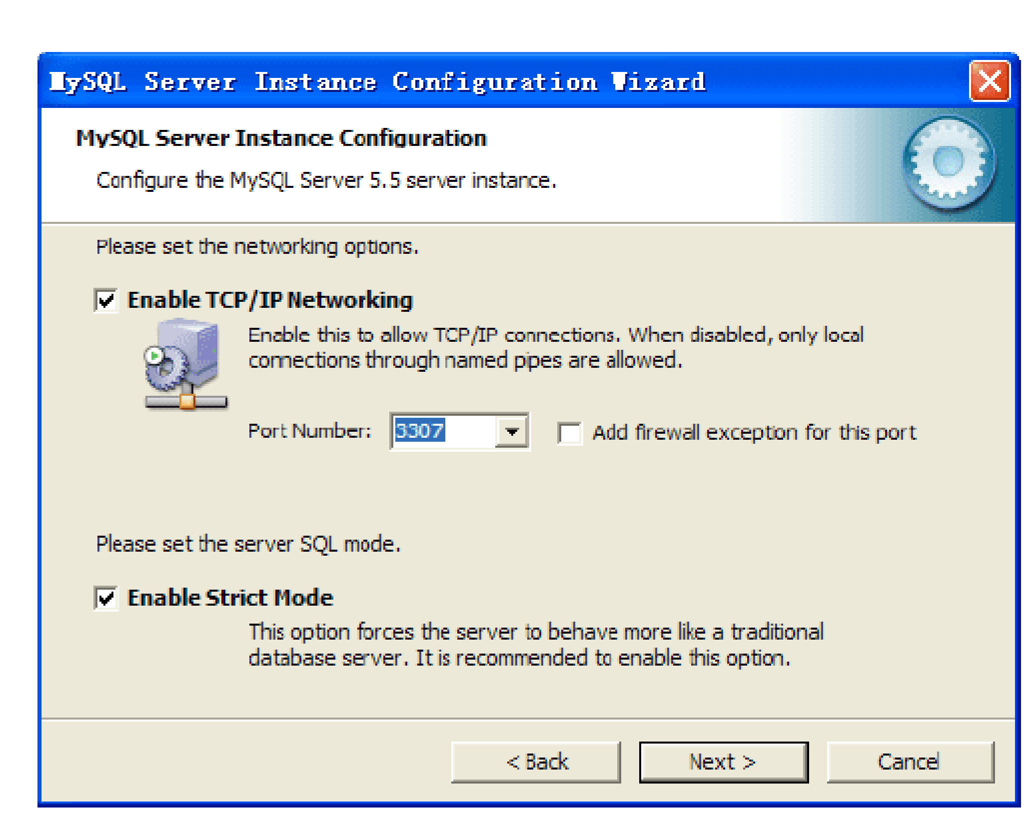


图 9-16　选择数据库端口号对话框

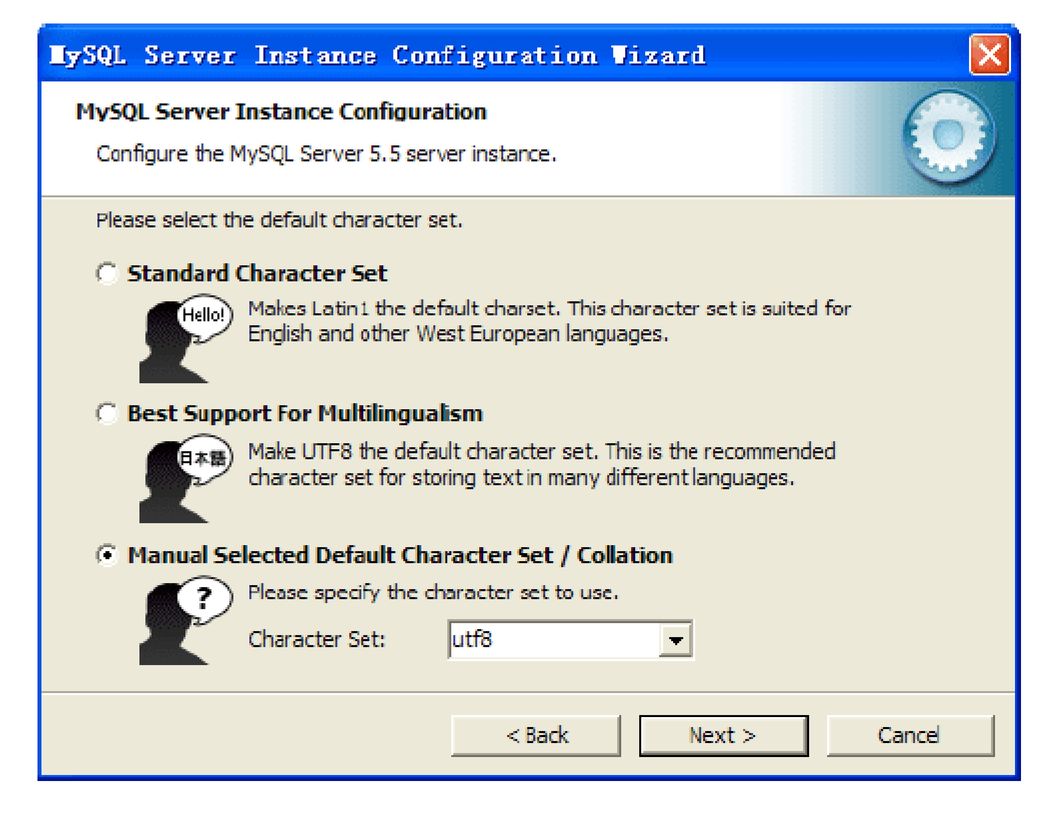


图 9-17　选择数据库使用的字符集

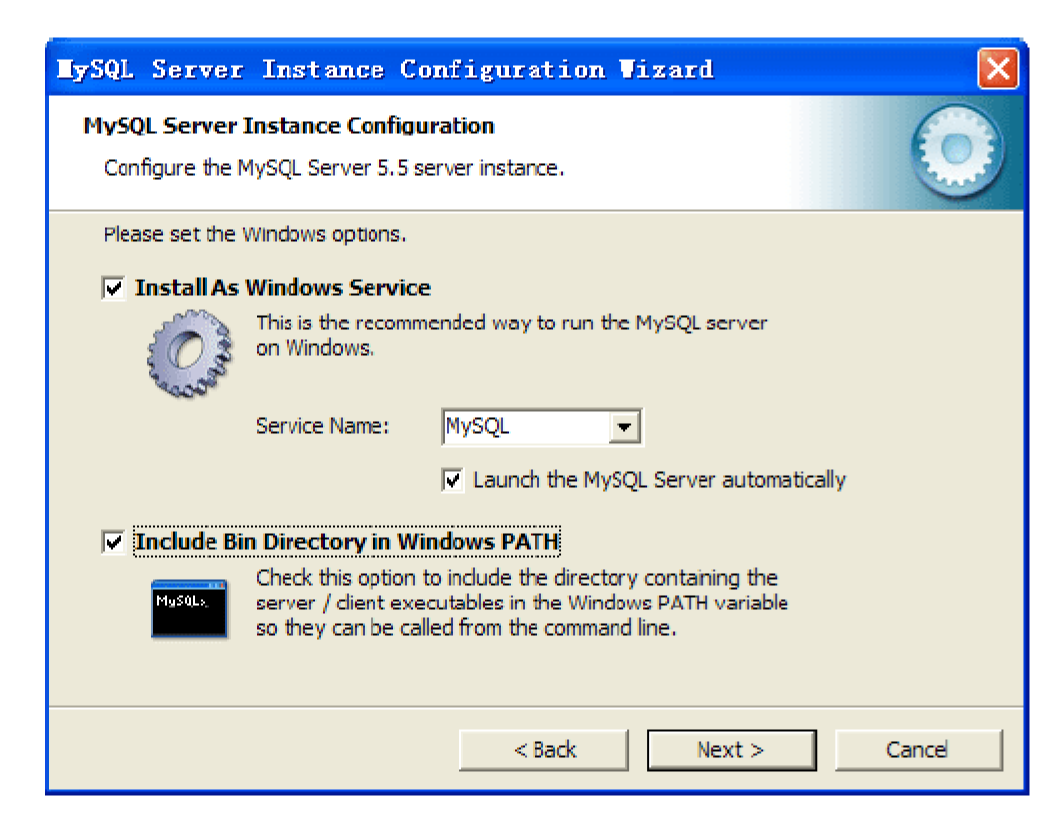


图 9-18　配置 Windows 选项

(15) 单击配置 Windows 选项中的 Next 按钮，弹出为 MySQL 数据库设置密码的对话框，如图 9-19 所示。设置完密码之后，单击 Next 按钮，弹出保存 MySQL 设置对话框，如图 9-20 所示。



图 9-19 为数据库设置密码

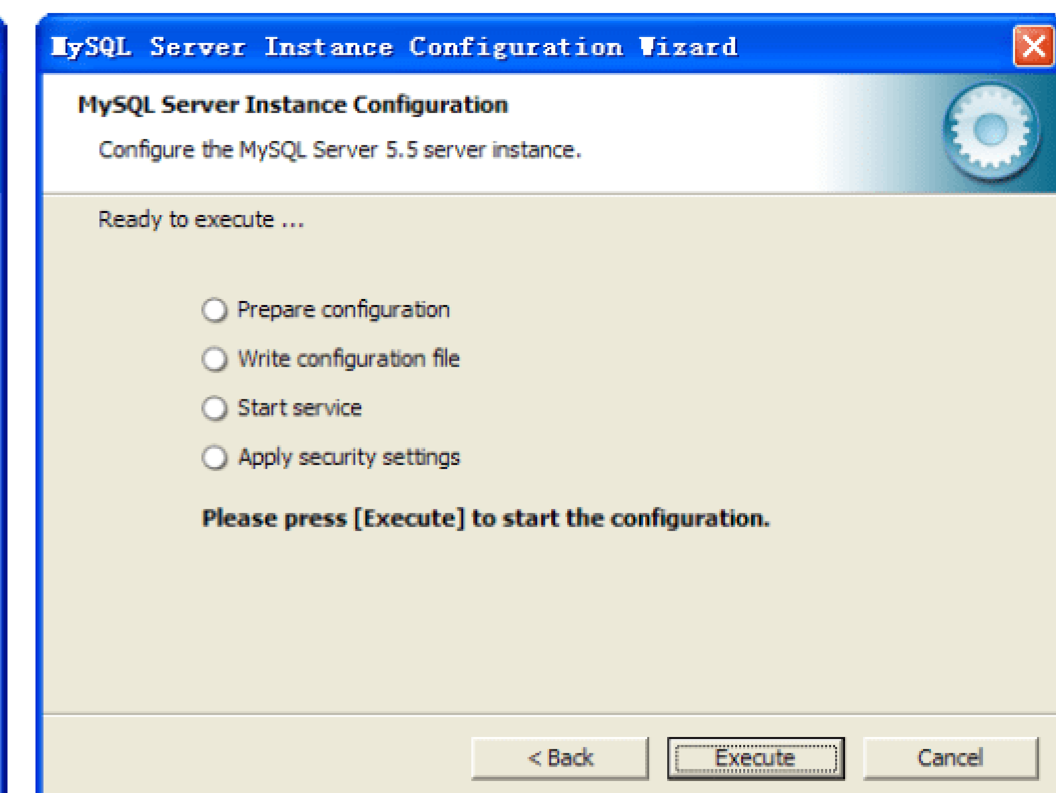


图 9-20 保存配置对话框

(16) 单击保存配置对话框中的 Execute 按钮，进行保存设置。如果保存成功，会弹出保存成功的对话框，最后单击 Finish 按钮，完成配置。

9.1.2 基础知识——安装 MySQL Administrator

通过上一节的讲述成功地安装了 MySQL 数据库，为了便于操作 MySQL，下面安装一个操作 MySQL 数据库的工具。在这里安装最常用的 MySQL Administrator，其详细安装步骤如下。

(1) 双击从网站下载好的 mysql-gui-tools-5.0-r17-win32.msi 文件，进入欢迎安装对话框，如图 9-21 所示。然后单击 Next 按钮，进入安装协议对话框，如图 9-22 所示。选择第一个单选按钮，该选项表示同意协议。



图 9-21 欢迎安装对话框

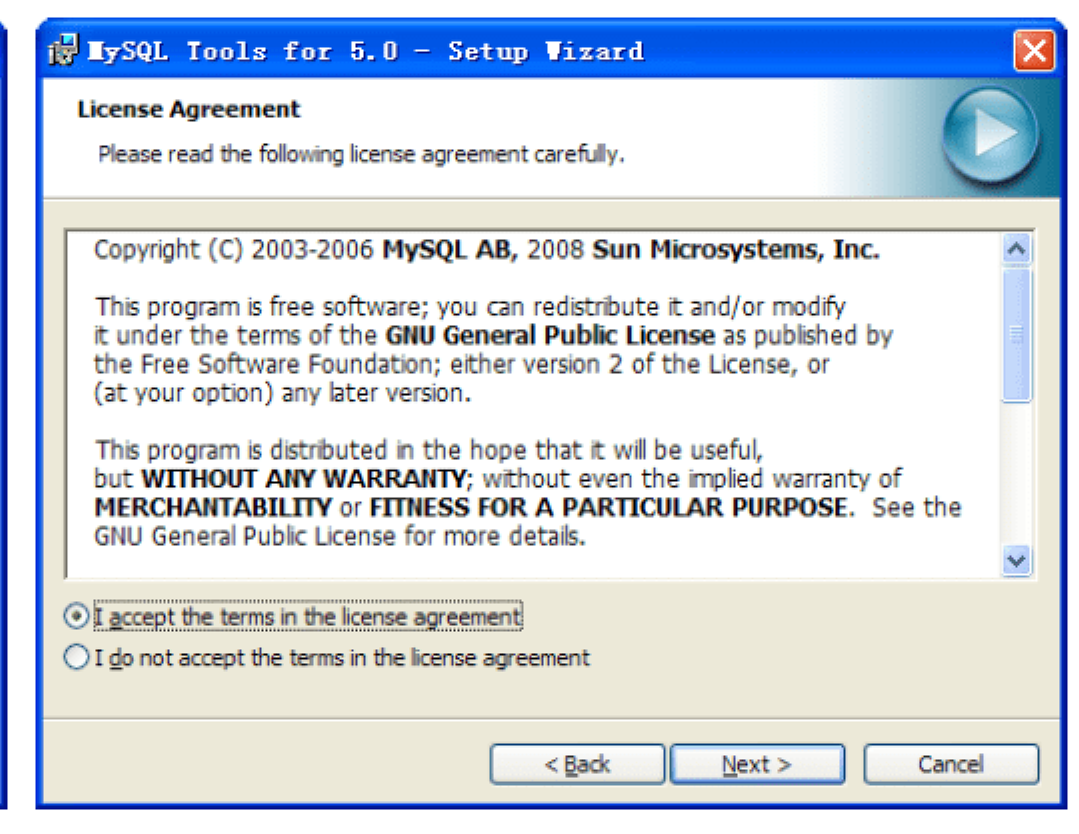


图 9-22 安装协议对话框

(2) 单击安装协议对话框中的 Next 按钮，进入选择路径对话框，如图 9-23 所示。选择安装路径后，单击 Next 按钮，进入选择安装类型对话框，如图 9-24 所示，这里使用默认值。

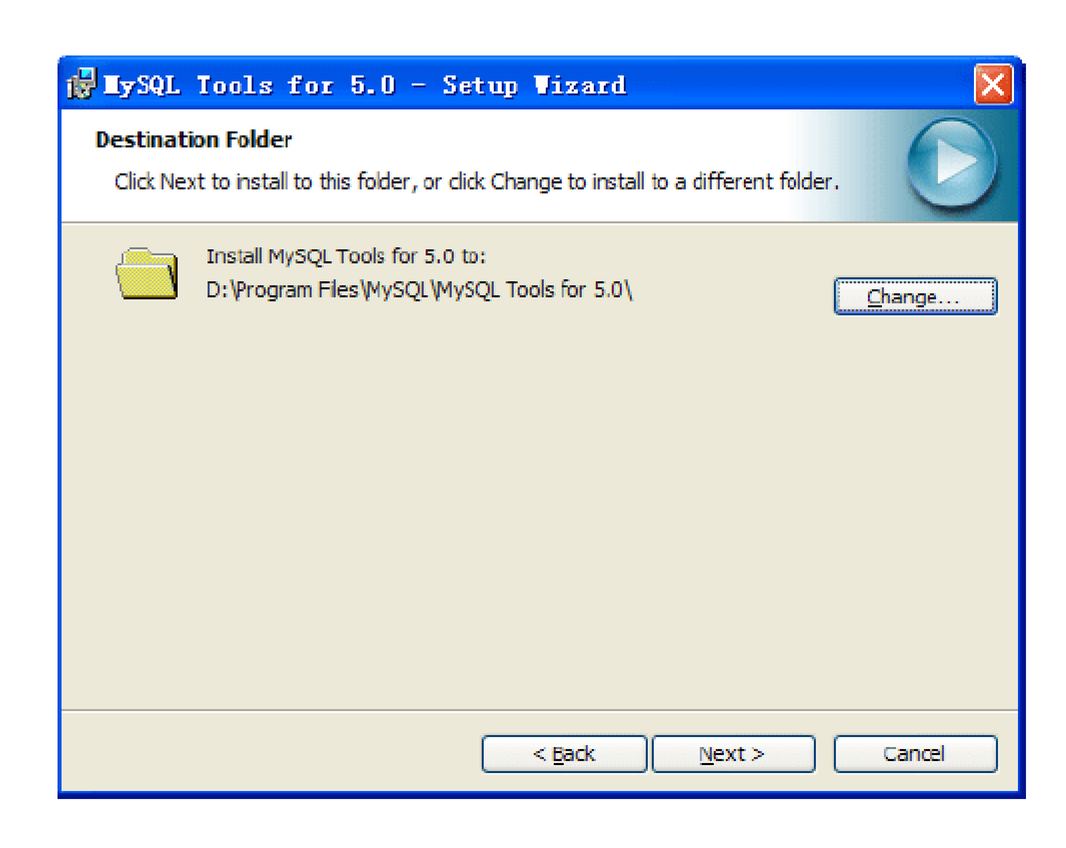


图 9-23　选择路径对话框

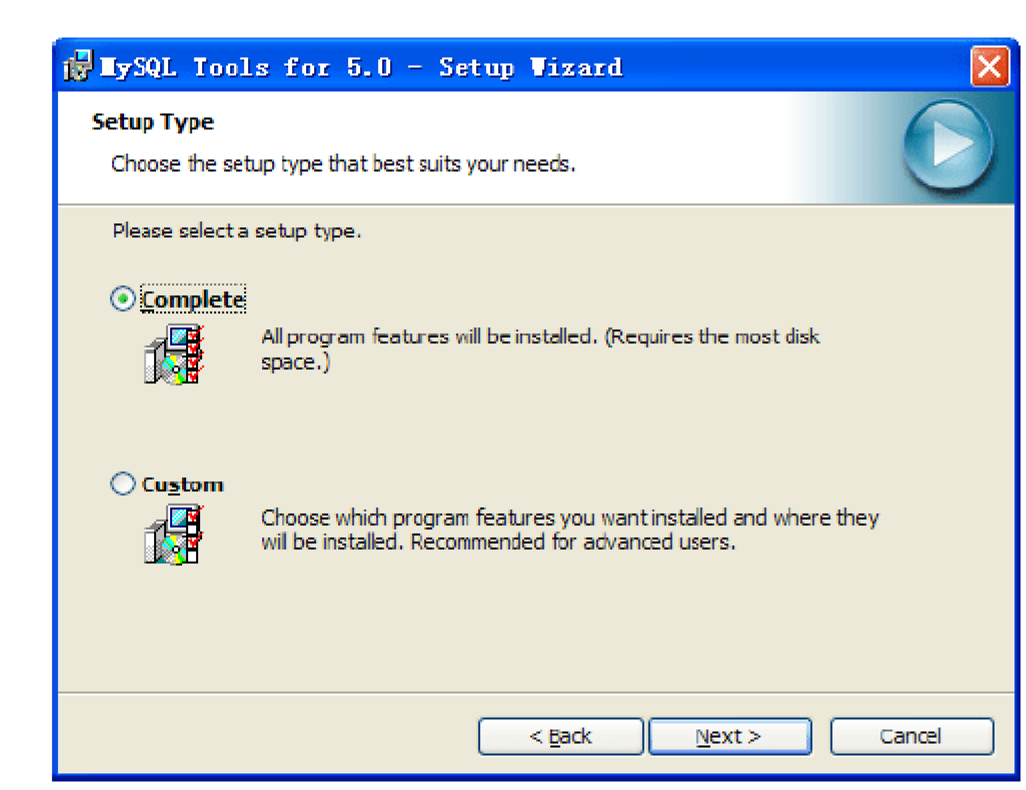


图 9-24　选择安装类型对话框

(3) 单击选择安装类型对话框中的 Next 按钮，进入准备安装工具对话框，如图 9-25 所示。单击 Install 按钮，弹出安装工具对话框，如图 9-26 所示。

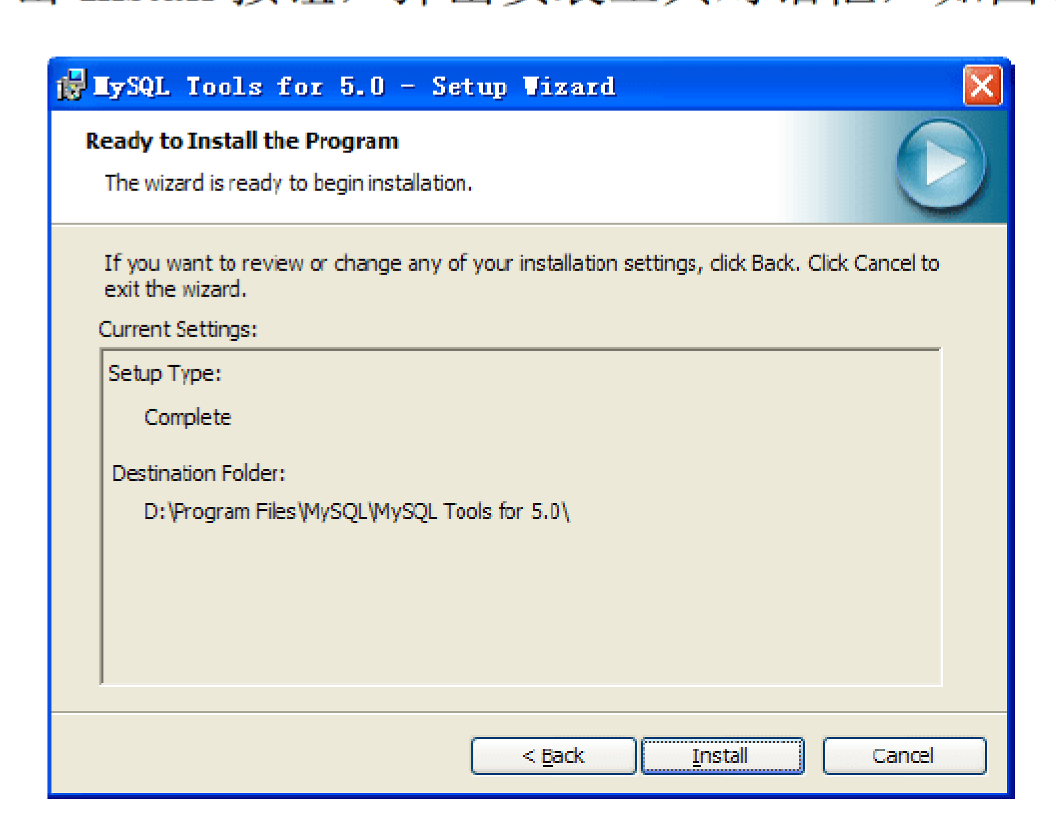


图 9-25　准备安装对话框

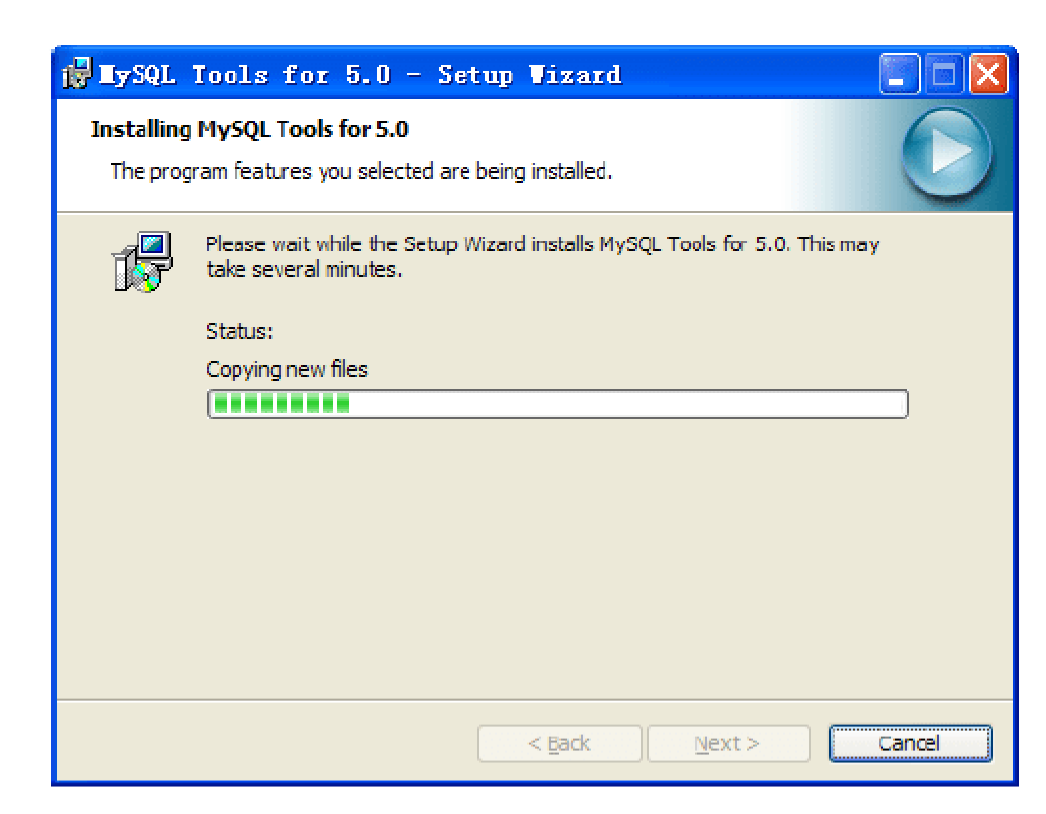


图 9-26　安装对话框

(4) 安装成功后，会弹出完成安装对话框，如图 9-27 所示。最后单击 Finish 按钮，完成安装。



图 9-27　安装完成对话框

9.1.3 基础知识——设置 MySQL 管理员密码

使用前面描述的安装 MySQL 配置向导，可以设置 MySQL 的密码，如果没有设置管理员密码，这将是非常不安全的事。因此，必须设一个密码，为此可以使用 SET PASSWORD 命令，如下：

```
%>mysql -u root mysql
%>SET PASSWORD FOR root@localhost=PASSWORD('123');
```

当然，要选择比“123”更复杂的密码，像 000、456、abc 之类的密码，MySQL 也会接受，但这是非常不安全的，应该考虑至少 8 个字符长、由数字和各种大小写的字母字符组成的密码。

如果不遵循立即选择密码的建议，意味着任何能够访问操作系统的人都可以关闭守护进程，甚至完全摧毁数据库服务器和数据，对系统而言将会造成很大的损失。

9.2 连接 MySQL 数据库

在 PHP 中，支持对多种数据库的操作，且提供了相关的数据库连接函数或操作函数。特别是与 MySQL 数据库的组合，PHP 提供了强大的数据库操作函数，可以直接在 PHP 中使用这些函数进行数据库的操作。操作数据库，首先需要进行数据库的连接，然后选择需要进行操作的数据库，再执行相关的数据库操作，最后需要关闭所建立的数据库连接。这一节主要讲解与数据库连接、关闭连接和选择数据库相关的函数。

**视频教学：**光盘\videos\09\登录到数据库.avi　　长度：3 分钟

9.2.1 基础知识——建立连接与关闭连接

在 PHP 中，要对数据库进行操作，首先需要与数据库建立连接，与数据库建立连接的函数有两个，分别是 mysql_connect()函数和 mysql_pconnection()函数。当对数据库操作完成后，需要关闭数据库连接，关闭数据库连接的函数为 mysql_close()函数。本章将对这 3 个函数做详细介绍。

1．打开数据库连接函数

在 PHP 中，使用 mysql_connect()函数打开一个到 MySQL 服务的非持久性的连接。mysql_connect()函数的语法格式如下：

```
resource mysql_connect ( [string server [, string user [, string pwd [, bool
newlink [, int clientflags]]]]] )
```

其中，参数 server 为要连接的数据库服务器的名称或 IP。参数 user 为连接数据库的用户名，若没有设置该参数，则默认为服务器进程所有者的用户名。参数 pwd 为连接数据库的密码，如果设置为该参数，则默认为空。

若每一次都使用同样的参数进行连接，在 PHP 中将不会进行新的数据库连接，而是直接

返回已经打开的数据库连接标识。参数 newlink 则改变了此行为，若设置该参数为布尔值 true，则将在每一次使用 mysql_connect()函数进行数据库连接时打开新的数据库连接，甚至在之前曾使用同样的参数进行过数据库的连接。

参数 clientflags 为设置客户端信息，它可以是以下常用的组合。

- MYSQL_CLIENT_COMPRESS：使用压缩协议。
- MYSQL_CLIENT_IGNORE_SPACE：允许在函数名后留空格位。
- MYSQL_CLIENT_INTERACTIVE：允许设置断开连接之前所空闲等候的 interactive_timeout 时间。
- MYSQL_CLIENT_SSL：使用 SSL 加密，本标志仅在 MySQL 客户端库版本为 4.x 或更高版本时可用。

该函数尝试打开或重复使用一个已打开的 MySQL 数据库服务器的连接。若成功连接，则返回一个 MySQL 连接标识，否则返回布尔值 false。这里的用户名(user)可以为超级管理员，也可以为用户表中存在的其他用户。如下面的语句将用超级管理员身份建立一个到本地服务器的连接：

```
$id=mysql_connect("localhost","root","000");
```

在上述的代码中使用 mysql_connect()函数连接本地 MySQL 数据库服务器，连接数据库的用户名为 root，密码为 000。若数据库服务器不可用，或连接数据库的用户名或密码错误，则可能引起一条 PHP 警告信息，如下所示：

```
Warning: mysql_connect() [function.mysql-connect]: Unknown MySQL server host
[...]
```



脚本一结束，到服务器的连接就被关闭，除非之前已经明确调用 mysql_close()关闭了。要创建一个持久连接，请使用 mysql_pconnect()函数。

下面创建一个实例，具体讲解如何使用 mysql_connect()函数与数据库建立连接，如下面的代码所示：

```
<?php
$con = mysql_connect("localhost","mysql_user","mysql_pwd");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
// 一些代码...
mysql_close($con);
?>
```

2. 建立持久性连接函数

在 PHP 中，使用 mysql_pconnect()函数打开一个到 MySQL 服务器的持久性连接，语法格式如下：

```
resource mysql_pconnect ( [string server [, string username [, string password
[, int client_flags]]]] )
```

该函数如果执行成功，则返回一个正的 MySQL 持久性标识符，否则返回 false。mysql_pconnect()建立一个到 MySQL 服务器的连接。如果没有提供可选参数，则使用如下默认值：server='localhost:3306'，username=服务器进程所有者的用户名，password=空密码。client_flags 参数可以是以下常量的组合：

- MYSQL_CLIENT_COMPRESS
- MYSQL_CLIENT_IGNORE_SPACE
- MYSQL_CLIENT_INTERACTIVE



server 参数也可以包括端口号，例如"hostname:port"，或者是本机套接字的路径，例如":/path/to/socket"。

mysql_pconnect()和 mysql_connect()非常相似，但有两个主要区别。

- 当连接的时候，此函数将先尝试寻找一个在同一主机上用同样的用户名和密码已经打开的持久连接，如果找到，则返回此连接标识而不打开新连接。
- 当脚本执行完毕后到 SQL 服务器的连接不会被关闭，此连接将保持打开以备以后使用，mysql_close()不会关闭由 mysql_pconnect()建立的连接。

下面使用该函数创建一个打开持久性连接的示例，代码如下：

```
<?php
    $con = mysql_pconnect("localhost:3306","root","000");
    if (!$con)
    {
        die('Could not connect: ' . mysql_error());
    }
?>
```

3. 关闭连接函数

在 PHP 中，使用 mysql_close()函数关闭 MySQL 连接，该函数的语法格式如下：

```
bool mysql_close ( [resource link_identifier] )
```

mysql_close()关闭指定的连接标识所关联的到 MySQL 服务器的非持久连接。如果没有指定 link_identifier，则关闭一个打开的连接。通常不需要使用 mysql_close()，因为已打开的非持久连接会在脚本执行完毕后自动关闭。该函数的参数 link_identifier 表示 MySQL 的连接标识符。MySQL 的连接标识符，如果没有指定，默认使用最后被 mysql_connect()打开的连接。如果没有找到该连接，函数会尝试调用 mysql_connect()建立连接并使用它。如果发生意外，没有找到连接或无法建立连接，系统发出 E_WARNING 级别的警告信息。

下面创建一个实例，具体说明如何使用 mysql_close()函数关闭连接，代码如下：

```
<?php
    $link = mysql_connect('localhost', 'mysql_user', 'mysql_password');
    if (!$link) {
        die('Could not connect: ' . mysql_error());
    }
    echo 'Connected successfully';
    mysql_close($link);
?>
```

9.2.2　基础知识——单独存放连接文件

在使用 PHP 语言建设网站的时候，避免不了多次连接数据库对其进行各种的操作，而在每次操作之前都需要打开与数据库的连接，那么如果在每次对数据库进行操作之前都编辑同样的打开数据库连接的代码，就会出现代码冗余的现象，同时会显得很繁琐。这种情况的解决办法就是将数据库连接的代码放到一个单独的文件中，其他的文件如果涉及到这个功能可以直接调用该文件。这样做的好处是方便数据库的代码修改，并达到功能相互分离和代码重用的效果。下面创建一个实例，独立 PHP 文件中的代码如下：

```
<?php
    $link=mysql_connect("localhost:3306","root","000");
?>
```

将该文件保存，名称为 sql_connection.php。如果一个 PHP 页面用到连接数据库的代码，可以通过下面的代码调用：

```
<?php
    include_once("sql_connection.php");
?>
```

9.2.3　基础知识——选择数据库

在进行数据库的连接后，需要在 PHP 脚本中选择需要进行操作的 MySQL 数据库。选择数据库可使用 mysql_select_db()函数，其语法格式如下：

```
bool mysql_select_db ( string database_name [, resource link_identifier] )
```

其中，参数 database_name 为要进行操作的数据库，参数 link_identifier 为创建的数据库连接标识。如果没有指定数据库连接标识，则使用上一个已打开的数据库连接标识；若没有已打开的数据库连接标识，函数将尝试使用无参数调用 mysql_connect()函数创建数据库连接并使用。函数尝试设定与指定的连接标识符所关联的 MySQL 数据库服务器上的当前激活数据库。若操作成功，则返回布尔值 true，否则返回 false。下面创建一个示例，具体介绍如何运用该函数选择数据库，代码如下：

```
<?php
    $con = mysql_connect("localhost:3306", "root", "000");
    if (!$con)
    {
        die('Could not connect: ' . mysql_error());
    }
    $db_selected = mysql_select_db("userDB", $con);
    if (!$db_selected)
    {
        die ("Can\'t use userDB : " . mysql_error());
    }
```

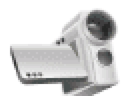
```
    echo "选择数据库成功！";
    mysql_close($con);
?>
```

在上段代码中，“userDB”表示在 MySQL 服务器中创建好的数据库。mysql_error()函数返回上一个 MySQL 操作产生的文本错误信息。当选择了数据库后，使用 mysql_close()函数关闭当前的连接。在浏览器中查看运行结果，在页面中输出。

```
选择数据库成功！
```

9.3 连接 MySQL 并实现添加学生信息功能

在上面的一节中讲解了如何连接 MySQL，以及如何选择数据库，而这些并未对数据库操作。进行数据库的操作，在 PHP 中需要使用一些函数来执行 MySQL 指令，本节主要讲解在 PHP 中执行 SQL 语句的函数。

 **视频教学：**光盘\videos\09\MySQL 基本命令.avi　　长度：7 分钟

9.3.1 基础知识——执行 SQL 语句

无论是向数据库中增添一条记录、删除记录、修改记录还是查询记录，都可以通过使用 mysql_query()函数和 mysql_db_query()函数将 SQL 语句传递给 MySQL 数据库管理系统，然后数据库管理系统执行这些 SQL 语句，执行完毕之后返回结果。本节主要讲解这两个函数如何在 PHP 中执行数据库操作。

1．mysql_query()函数

进行数据库的操作，在 PHP 中需要使用一个函数来执行 MySQL 指令，这就是 mysql_query()函数。其语法格式如下所示：

```
resource mysql_query ( string query [, resource link_identifier] )
```

其中，参数 query 为要执行的 MySQL 语句，参数 link_identifier 为打开的数据库连接。若没有设置参数 link_identifier，则函数使用上一个已经打开的数据库连接；如果没有已经打开的数据库连接，该函数将尝试以无参数方式调用 mysql_connect()函数，创建一个数据库连接并使用。

函数向参数 link_identifier 所关联的数据库服务器中的当前活动数据库发送一条查询语句。函数仅对 SELECT、SHOW、EXPLAIN 或 DESCRIBE 语句返回一个资源标识符，用于存储 SQL 语句的执行结果，若执行不成功则返回 false。对于其他类型的 SQL 语句，函数在执行成功时返回 true，出错时返回 false。因此，false 的返回值意味着此次查询合法并能够被 SQL 服务器成功执行，但并不意味着影响或者返回的行数，因为有可能此次 SQL 语句成功执行，但是并没有影响到或返回任何一行。

下面创建一个实例，执行一条创建数据库的语句，代码如下所示。

```
<?php
    mysql_connect("localhost","root","000") or die("数据库连接失败！");
    //连接 MySQL 数据库
    mysql_select_db("db_student") or die("选择的数据库不存在或不可用！");
    //选择数据库
    $myquery=mysql_query("select * from student") or die("SQL 语句执行失败！");
    //执行 SQL 语句
    mysql_close;
    //关闭数据库连接
?>
```

在上述代码中，进行了一次完整的数据库连接操作。先连接数据库服务器，然后选择数据库，执行 SQL 语句，最后关闭数据库连接，使用了 mysql_query()函数执行 SQL 语句。

2. mysql_db_query()函数

除了 mysql_query()函数能执行 SQL 语句外，PHP 还提供了另一函数，即 mysql_db_query()函数。该函数与 mysql_db_query()函数和 mysql_query()函数的功能类似，其区别在于 mysql_db_query()函数在执行 SQL 语句时可以同时选择数据库。其语法格式如下所示。

```
resource mysql_db_query ( string database, string query [, resource
link_identifier] )
```

其中，参数 database 为要执行 SQL 语句的数据库，参数 query 为要执行的 SQL 语句，参数 link_identifier 为数据库连接标识符。若没有设置该参数，函数将打开上一个已经创建的数据库连接；若找不到已打开的数据库连接，则尝试以无参数方式调用 mysql_connect()函数创建一个数据库连接。

函数在执行成功时，根据查询结果返回一个 MySQL 结果资源号，出错时将返回 false。函数会对 INSERT、UPDATE 和 DELETE 语句返回 true 或 false 来指示执行成功或失败。



在使用该函数后，不会自动切换回先前使用 mysql_select_db()函数连接的数据库。若需要再次使用先前连接的数据库，需要再次手动指定，因此建议在查询时使用 database.table 的格式来替换该函数。

下面创建一个使用该函数的示例，代码如下所示。

```
<?php
    mysql_connect("localhost","root","000") or die("数据库连接失败！");
                    //连接 MySQL 数据库
    $myquery=mysql_db_query("db_student","select * from student") or die("SQL
语句执行失败！");      //采用 mysql_db_query()函数进行查询
    mysql_close();
?>
```

在上述代码中，使用 mysql_db_query()函数进行数据库的查询。因为可以在该函数内指定要查询的数据库，所以在以上脚本中不再指定要查询的数据库。

9.3.2 实例描述

学生信息管理系统中不可缺少的功能就是添加学生信息到数据库表中。当新增一个学生成员时，就需要向数据库表中添加一行新数据，这条数据包括学生的学号、姓名、年龄、性别和联系地址。下面我们就来使用 MySQL 的 mysql_select_db()函数和 mysql_query()函数实现这个功能吧！

9.3.3 实例应用

【例 9-1】 连接 MySQL 并实现添加学生信息功能

(1) 新建 PHP 文件，命名为 9-3.php。

(2) 9-3.php 页面是输入新增学生信息页面，在该页面中创建表单域，代码如下。

```
<form name="myForm" method="post" action="doInsert.php">
    <table   cellpadding="0"   cellspacing="0"   border="1"   width="60%"
style="margin-top:-50px;">
        <tr>
            <td>学号</td>
            <td><input type="text" name="txtId"  /></td>
        </tr>
        <tr>
            <td>姓名</td>
            <td><input type="text" name="txtName" /></td>
        </tr>
        <tr>
            <td>年龄</td>
            <td><input type="text" name="txtAge"></td>
        </tr>
        <tr>
            <td>性别</td>
            <td>
                <select name="sSex">
                    <option value="male">male</option>
                    <option value="female">female</option>
                </select>
            </td>
        </tr>
        <tr>
            <td>地址</td>
            <td><input type="text" name="txtAddress"/></td>
        </tr>
        <tr>
            <td colspan="2" align="center">
                <input type="submit" name="submit" value="添加"/>
                <input type="reset" name="reset" value="重置">
            </td>
        </tr>
    </table>
</form>
```

(3) 当用户输入新增学生信息后，单击【添加】按钮，表单提交至 doInsert.php 页面，在该页面中编辑 PHP 脚本代码，执行向数据库表中添加数据操作。该项目使用的数据库名为 db_student，表名为 student。执行向 student 表中添加数据的 PHP 脚本代码如下：

```
<?php
    //连接 MySQL 数据库
    $link=mysql_connect("localhost:3306","root","000");
    //选择数据库
    mysql_select_db("db_student");
    //获取表单提交过来的数据
    $id=$_POST['txtId']|null;
    $name=$_POST['txtName'];
    $age=$_POST['txtAge'];
    $sex=$_POST['sSex'];
    $address=$_POST['txtAddress'];
    //添加数据的 SQL 语句
    $exec="insert into student values($id,'$name',$age,'$sex','$address')";
    //执行添加操作，返回受影响的行数
    $result=mysql_query($exec);
    if($result) echo "学生已添加到数据表中<br>";
    else echo "该学生没有添加进数据表，请重新输入。<br>";
    mysql_close();
?>
```

(4) 编辑 header.htm 和 menu.htm，分别作为 Frameset 框架的顶部页面和左边菜单导航页面。

(5) 编辑 HTML 文件，命名为 9-3-index.htm。

(6) 该页面使用 Frameset 框架，将 header.htm、menu.htm 和 9-3.php 整合。

9.3.4　运行结果

在浏览器中运行 9-3-index.htm 页面，在页面中显示输入新增学生信息的表单域，如图 9-28 所示。



图 9-28　输入新增学生信息页面

当用户输入新增学生信息后，单击【添加】按钮，提交至 doInsert.php 页面，处理表单提交数据，并添加到数据库表中。如果添加成功，在页面中输出成功信息，如图 9-29 所示。



图 9-29　添加成功页面

9.3.5　实例分析



源码解析：

在上述案例中的 9-3.php 页面中，创建一个输入新增学生信息的表单域，当用户输入学生学号、姓名、年龄、性别和住址后，表单提交至 doInsert.php 页面。在该页面中，首先使用 mysql_connect()函数获取与数据库的连接，接着使用$_POST 获取表单元素值，然后编写插入的 SQL 语句，将获取到的用户输入值添加到数据库表 student 中，并使用 mysql_query()函数执行添加操作。最后判断 mysql_query()函数返回的结果是否是 true，如果是，在页面中输出“学生已添加到数据表中”，否则，输出“该学生没有添加进数据表，请重新输入”。

9.4　查询并显示学生信息数据

在前面的一节中已经讲述了如何使用 MySQL 函数执行 SQL 语句，以实现对数据库的操作。本节将讲解如何把执行函数获得的资源标识符对象显示在页面中。

 **视频教学：**光盘\videos\09\分析结果集.avi　　　　长度：10 分钟

9.4.1　基础知识——分析结果集

在每一次成功执行 SQL 语句后，mysql_query()函数总是会返回一个结果集。要对结果集进行分析，首先需要获取所执行的 SQL 语句影响的行数。

1．获取结果集中包含的记录数

对于该结果集中包含的记录数，可使用 mysql_num_rows()函数获取。其语法格式如下所示：

```
int mysql_num_rows ( resource result )
```

其中，参数 result 为函数 mysql_query()所返回的结果集。函数返回结果集中的记录数。下面创建一个示例，具体介绍如何在 PHP 脚本中获取结果集中的记录数，代码如下所示：

```
<?php
    mysql_connect("localhost","root","000") or die("数据库连接失败！");
    //连接数据库服务器
    mysql_select_db("db_student") or die("选择的数据库不存在或不可用！");
    //选择数据库
    $myquery=mysql_query("select * from student") or die("SQL 语句执行失败！");
    //执行 SQL 语句
    echo "结果集中的行数为：".mysql_num_rows($myquery);//获取结果集中的记录数
    mysql_close();
?>
```

在上述代码中，使用 mysql_num_rows()函数获取结果集中的记录数。运行该段代码，在页面中输出结果如下所示：

```
结果集中的行数为：2
```

2．获取受影响行数

若在 mysql_query()函数中使用 INSERT、UPDATE 和 DELETE 语句，应使用 mysql_affected_rows()函数获取所影响到的记录数，其语法格式如下所示：

```
int mysql_affected_rows ( [resource link_identifier] )
```

其中，参数 link_identifier 为已打开的数据库连接标识符。若未设置该参数，函数默认使用上一次所打开的数据库连接；若未找到该连接，函数将尝试以无参数方式调用 mysql_connect()函数建立数据库连接并使用；若发生意外，如找不到数据库连接或创建数据库连接失败时，将产生一条警告信息。

函数返回由参数 link_identifier 所关联的数据库连接进行的 INSERT、UPDATE 和 DELETE 查询所影响到的行数。函数执行成功则返回最后一次操作所影响的函数，若最后一次查询失败，则返回-1。在使用 UPDATE 语句时，MySQL 不会将原值与新值一样的列进行更新，因此该函数所返回的值不一定就是使用 mysql_query()函数所影响到的行数，此时只是返回真正被更新的行数。

3．获取结果集中的数据

要显示对于使用 mysql_query()函数所返回的结果集，首先需获取结果集中的数据。获取结果集中的某一条数据可使用 mysql_result()函数，其语法格式如下所示：

```
mixed mysql_result ( resource result, int row [, mixed field] )
```

mysql_result()函数有 3 个参数。

- data：该参数是必需的，规定要使用的结果标识符，该标识符是 mysql_query()函数返回的。
- row：该参数是必需的，规定行号，行号从 0 开始。
- field：该参数是可选的，规定获取哪个字段，可以是字段偏移值，字段名或 table.fieldname。如果该参数未规定，则该函数从指定的行获取第一个字段。

下面创建一个示例，具体讲解如何使用该函数返回结果集中的某一个字段的值，代码如下所示：

```
<?php
    mysql_connect("localhost:3306", "root", "000") or die("数据库连接失败！");
    //连接数据库服务器
    $db_selected = mysql_select_db("user") or die("选择的数据库不存在或不可用");
    //选择数据库
    mysql_query("set names gb2312");//设置输出编码
    $sql = "SELECT * from tab_user";
    //执行 SQL 语句
    $result = mysql_query($sql) or die("SQL 语句执行失败！");
    //显示结果集中第一行数据
    echo "用户名为：".mysql_result($result,0,"username")."<br/>";
    //显示结果集中第二行数据
    echo "真实姓名为：".mysql_result($result,0,"realname")."<br/>";
    //显示结果集中第三行数据
    echo "密码为：".mysql_result($result,0,"password");
    mysql_close($con);
?>
```

在上述代码中，使用 mysql_result()函数获取结果集中第一行数据信息。保存上段代码并执行，运行结果如下：

```
用户名为：yinguopeng
真实姓名为：YinPeng
密码为：mxlygp
```



因 MySQL 服务器中存储数据的编码与浏览器显示结果的编码有可能不一致，所以在程序中使用了“set names gb2312”直接指定其内容输出的编码。若未指定输出字符编码，则可能出现乱码。

以上实现了获取结果集中的某一行数据。其实通过与 mysql_num_rows()函数所返回的行数，可以输出结果集中的所有数据，代码如下：

```
<?php
    mysql_connect("localhost", "root", "000") or die("数据库连接失败！");
    //连接数据库服务器
    $db_selected = mysql_select_db("user") or die("选择的数据库不存在或不可用");
    //选择数据库
    mysql_query("set names gb2312");
    //设置输出编码
    $sql = "SELECT * from tab_user";
```

```
    //执行 SQL 语句
    $result = mysql_query($sql) or die("SQL 语句执行失败！");
    $rowscnt=mysql_num_rows($result);
    //获取结果集行数
    echo "<table border=1><tr><td>编号</td><td>用户名</td><td>真实姓名</td><td>
密码</td></tr>";
    for($i=0;$i<$rowscnt;$i++)
    {
        echo "<tr><td>".mysql_result($result,$i,0)."</td>";
        echo "<td>".mysql_result($result,$i,1)."</td>";
        echo "<td>".mysql_result($result,$i,2)."</td>";
        echo "<td>".mysql_result($result,$i,3)."</td></tr>";
    }
    echo "</table>";
    mysql_close();
?>
```

在上述程序中，通过对结果集进行循环输出结果集中的所有数据信息。其输出结果如图 9-30 所示。

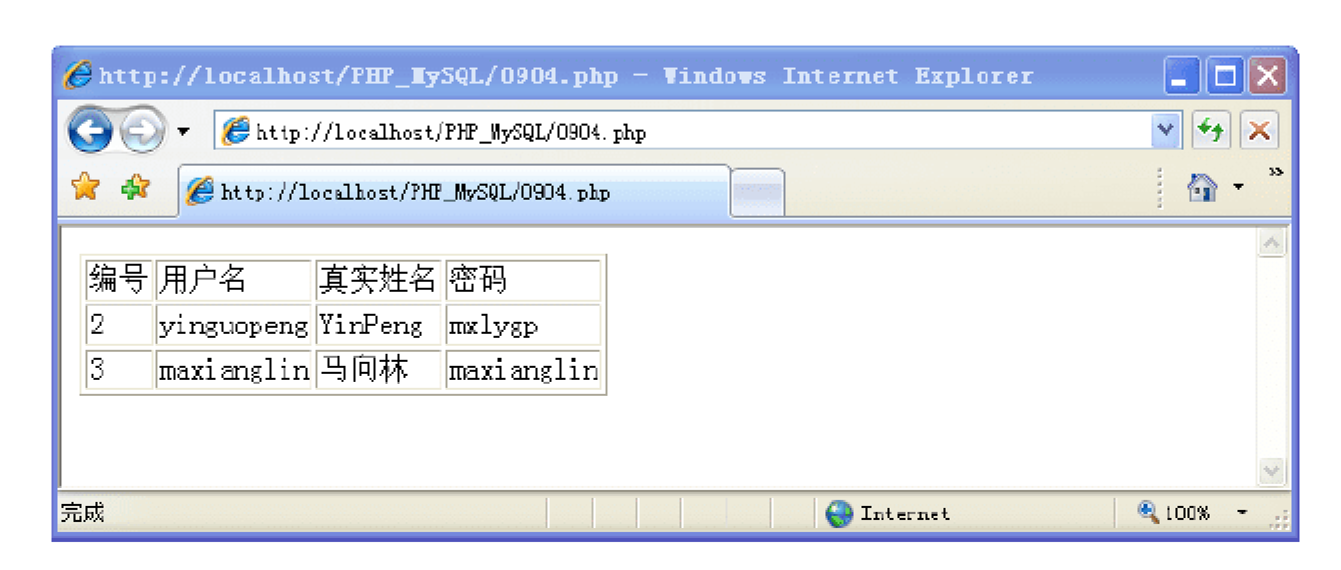


图 9-30　输出全部数据

4．逐行获取结果集中的记录

当需要在结果集中获取记录时，使用 mysql_result()函数就显得有些复杂，并且处理效率也低了很多。为此，PHP 提供了从结果集中获取逐行记录的函数 mysql_fetch_row()，其语法格式如下所示：

```
array mysql_fetch_row ( resource result )
```

其中，参数 result 为 mysql_query()函数所返回的结果集。该函数从指定的结果集中取得一行数据并以数组的形式返回。数组以列的顺序依次进行分配。依次调用该函数将返回结果中的下一行，若没有更多时则返回 false。

下面创建一个使用该函数的实例，代码如下：

```
<?php
    $link=mysql_connect("localhost:3306","root","000") or die("数据库连接失败！
");
    //选择数据库
    mysql_select_db("db_student");
    $sqlText="select * from student";
```

```
    //执行 SQL 语句
    $result=mysql_query($sqlText);
    echo "<fieldset><legend>从数据库中获取的数据</legend>";
    echo "<table width='100%'><tr><td>学号</td><td>姓名</td><td>年龄</td><td>
性别</td><td>地址</td></tr>";
    while (list($id,$name,$age,$sex,$address)=mysql_fetch_row($result))
    {
        echo "<tr><td>".$id."</td><td>".$name."</td><td>".$age."</td><td>";
        if($sex=="female")
        {
            echo "女";
        }
        else
        {
            echo "男";
        }
        echo "</td><td>".$address."</td></tr>";

    }
    echo "</table></fieldset>";
    mysql_close();

?>
```

保存上段代码，在浏览器中查看运行结果，如图 9-31 所示。

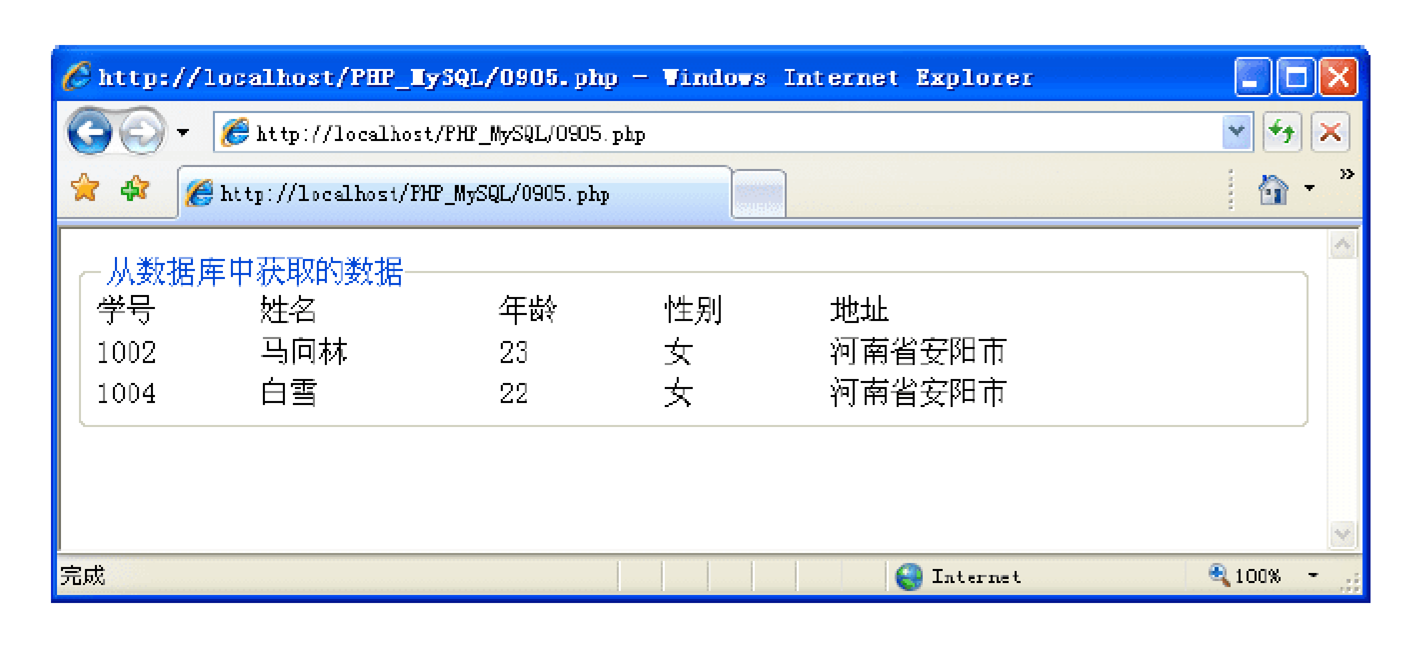


图 9-31　使用 mysql_fetch_row()函数

5．根据需要获取结果集中的数据

在函数 mysql_fetch_row()所返回的每一行结果数组中，该行的第一列在数组中是以 0 开始，所以在实际的应用中很不方便。PHP 提供了另一功能强大的函数 mysql_fetch_array()，其语法格式如下所示：

```
array mysql_fetch_array ( resource result [, int result_type] )
```

该函数有两个参数，data 参数为可选的参数，规定要使用的数据指针，该数据指针是 mysql_query()函数产生的结果；array_type 参数也为可选参数，规定返回哪种结果，可能的值有如下 3 个。

- MYSQL_ASSOC：返回根据结果集中的某一行所生成的关联数组。关联数组以数据库中的列名作为数组键名，其返回数组与函数 mysql_fetch_assoc()函数返回的结果一样。
- MYSQL_NUM：返回根据结果集中的某一行所生成的索引数组。索引数组中的键名以 0 开始，依次顺序分配，其返回结果数组与函数 mysql_fetch_row()返回的结果一样。
- MYSQL_BOTH：返回根据结果集中的某一行所生成的关联和索引数组，在未指定结果类型时默认为该值。

函数返回根据从结果集中取得的一行所生成的数组，若没有更多行时，返回 false。



该函数所返回的数组中列名是区分大小写的。

下面使用该函数创建一个示例，具体代码如下：

```
<?php
    mysql_connect("localhost","root","000") or die("数据库连接失败！");
        //连接数据库服务器
    mysql_select_db("db_student") or die("选择的数据库不存在或不可用！");
        //选择数据库
    mysql_query("set names gb2312");    //设置输出字符编码
    $sql=mysql_query("select * from student") or die("SQL 语句执行失败！");
        //执行 SQL 语句
    echo "<table border=1><tr><th>学号</th><th>姓名</th><th>年龄</th><th>性别
</th><th>住址</th></tr>";
    while ($row=mysql_fetch_array($sql,MYSQL_BOTH))
    {   //循环获取结果集中的数据
        echo "<tr><td>".$row[0]."</td>";
        echo "<td>".$row[1]."</td>";
        echo "<td>".$row[2]."</td>";
        echo "<td>".$row["sex"]."</td>";
        echo "<td>".$row[4]."</td></tr>";
    }
    echo "</table>";
    mysql_close();
?>
```

在浏览器中查看运行结果，效果如图 9-32 所示。



图 9-32　使用 mysql_fetch_array()函数效果图

在上述代码中，使用 mysql_fetch_array()函数循环获取结果集中的数据。通过这个例子可

以发现，程序输出结果与前例结果完全一样。因此该函数具有很强的灵活性，在实际应用中通常使用该函数。

当结果集中的数据非常多时，使用上述的方式会使页面很长，不美观，并且用户查询信息也很麻烦，因此需要对结果集进行分页显示。代码如下：

```
<?php
    mysql_connect("localhost","root","000") or die("数据库连接失败！");
    //连接数据库服务器
    mysql_select_db("db_student") or die("选择的数据库不存在或不可用！");
    //选择数据库
    mysql_query("set names gb2312");    //设置输出字符编码
    $sql=mysql_query("select * from student") or die("SQL 语句执行失败！");
    //执行 SQL 语句
    $page_size=5;                       //设置每页显示记录数
    $num_cnt=mysql_num_rows($sql);      //获取记录总数
    $page_cnt=ceil($num_cnt/$page_size);//计算总页数
    if(isset($_GET['p']))               //设置第一页还是其他的第几页
    {
        $page=$_GET['p'];
    }
    else {
        $page=1;
    }
    $query_start=($page-1)*$page_size;  //设置每页开始的记录号
    $pagesql="select * from student limit $query_start,$page_size";
    //使用 limit 获取记录
    $queryset=mysql_query($pagesql);    //执行 SQL 语句
    echo "<table border=1><tr><th>学号</th><th>姓名</th><th>年龄</th><th>性别
</th><th>住址</th></tr>";
    while ($row=mysql_fetch_array($queryset,MYSQL_BOTH))
    {  //循环获取结果集中的数据
        echo "<tr><td>".$row[0]."</td>";
        echo "<td>".$row[1]."</td>";
        echo "<td>".$row[2]."</td>";
        echo "<td>".$row["sex"]."</td>";
        echo "<td>".$row[4]."</td></tr>";
    }
    echo "</table>";
    $pager="共".$page_cnt."页 跳转至";    //显示分页部分
    if($page_cnt>1)                     //页面总数大于 1，则分页显示
    {
        for($i=1;$i<=$page_cnt;$i++)
        {
            if($page==$i)
            {
                $pager.="<a href='?p=$i'><b>$i</b></a>";
            }
            else
```

```
            {
                $pager.="<a href='?p=$i'>$i</a>";
            }

        }
        echo $pager."页";
    }
    echo $page;
    mysql_close();
?>
```

运行该段代码所在的页面 page.php，在浏览器中出现如图 9-33 所示的页面，当单击图中的“2”时，页面显示第 2 页的内容，如图 9-34 所示。

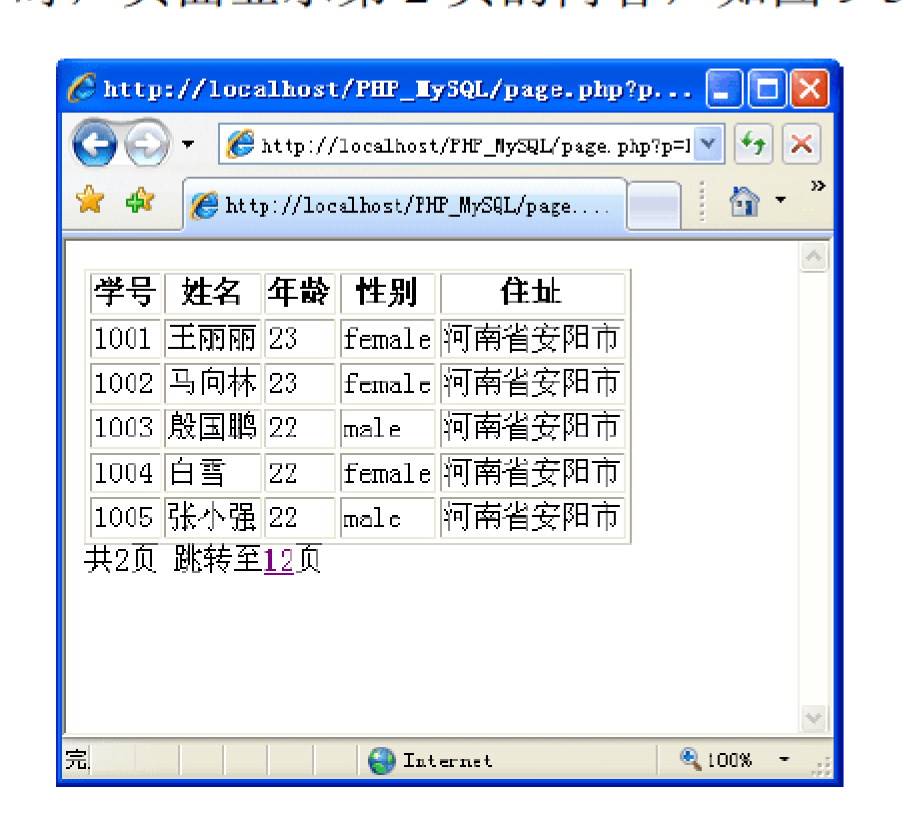



图 9-33　分页显示第 1 页内容



图 9-34　分页显示第 2 页内容

在上述程序中，分页显示结果集，每一页 5 条记录。首先获取结果集的总数，然后根据每一页显示的记录数计算总的页数，再由当前页数和记录数计算每一页的起止记录数，并使用 SELECT 语句的 LIMIT 子句实现，最后加粗显示当前页码。

6．mysql_fetch_assoc()函数

mysql_fetch_assoc()函数从结果集中取得一行作为关联数组，返回根据从结果集取得的行生成的关联数组，如果没有更多的行，则返回 false。该函数的语法格式如下：

```
array mysql_fetch_assoc ( resource result )
```



mysql_fetch_assoc()和用 mysql_fetch_array()加上第二个可选参数 MYSQL_ASSOC 完全相同。它仅仅返回关联数组，这也是 mysql_fetch_array() 初始的工作方式。

下面使用该函数创建一个示例，代码如下：

```
<?php
    $link=mysql_connect("localhost:3306","root","000");
    if(!$link)
    {
        echo "数据库连接失败！";
    }
    mysql_select_db("db_student");//选择数据库
```

```
    $sqlText="select * from student";
    $result=mysql_query($sqlText);//执行 SQL 语句
    while ($row=mysql_fetch_assoc($result)) {
        echo "编号: ".$row['id']."姓名: ".$row['name']."年龄: ".$row
['age']."性别: ".$row['sex']."住址: ".$row['address']."<br/>";

    }
?>
```

保存上段代码并执行，在页面中显示查询数据，如图 9-35 所示。

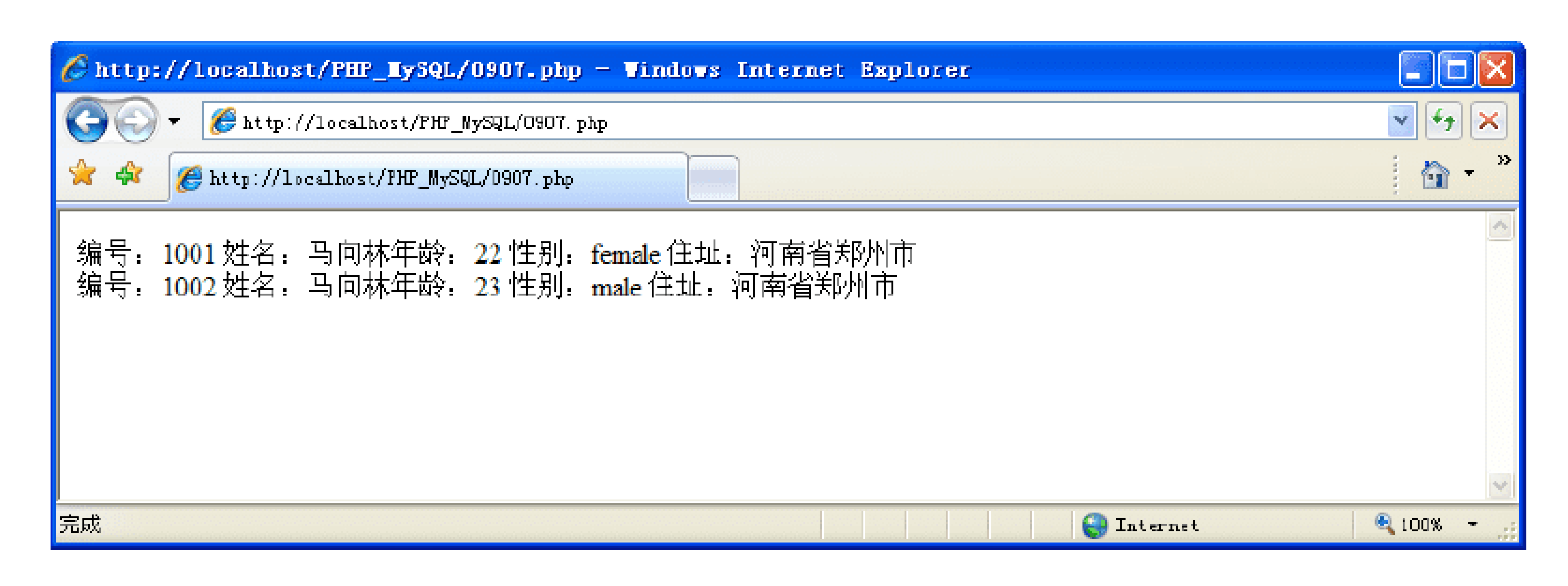


图 9-35　使用 mysql_fetch_assoc()函数

9.4.2　实例描述

在上一节的案例中讲述了 PHP 如何使用 MySQL 函数执行一条 SQL 语句，添加一条新增学生信息到数据库表中。如果只是单单的添加操作，无法查看到数据库中是否真的有该条数据，那么这个系统将是很失败的，因此，应当在上面案例的基础之上进行完善，当添加成功之后，查询数据库表中的所有信息并显示在页面中。

9.4.3　实例应用

【例 9-2】 查询并显示学生信息数据

(1) 新建 PHP 文件，命名为 9-4.php。

(2) 在 9-4.php 页面中编辑 PHP 脚本代码，获取与数据库的连接并执行查询操作，以表格的形式显示 student 表中的数据信息。代码如下。

```
<?php
    $db_host="localhost";   //MYSQL 服务器名
    $db_user="root";        //MYSQL 用户名
    $db_pass="000";         //MYSQL 用户对应密码
    $db_name="db_student";  //要操作的数据库
    //使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
    $link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器
".mysql_error());
    mysql_select_db($db_name,$link);    //选择相应的数据库，这里选择 db_student 库
    $sql="select * from student";       //先执行 SQL 语句显示所有记录以与插入后相比较
```

```
    $result=mysql_query($sql,$link);   //使用 mysql_query()发送 SQL 请求
    echo "<table cellSpacing=0 cellPadding=0 width='95%' align=center border=1
style='margin-top:-20px;border-color:#003366'>";         //使用表格格式化数据
    echo  "<tr><td  style='border-color:#003366'> 学 生 学 号 </td><td  style=
'border-color:#003366'>学生姓名</td><td style='border-color:#003366'>学生年龄
</td><td  style='border-color:#003366'> 学 生 性 别 </td><td  style='border-
color:#003366'>家庭住址</td></tr>";
    while($row=mysql_fetch_array($result)) //遍历 SQL 语句执行结果把值赋给数组
    {
        echo "<tr>";
        echo "<td style='border-color:#003366'>".$row[id]."</td>";   //显示学号
        echo "<td style='border-color:#003366'>".$row[name]." </td>";
        //显示姓名
        echo "<td style='border-color:#003366'>".$row[age]." </td>";
        //显示年龄
        echo "<td style='border-color:#003366'>".$row[sex]." </td>";
        //显示性别
        echo "<td style='border-color:#003366'>".$row[address]." </td>";
        //显示地址
         echo "</tr>";
    }
    echo "</table>";
?>
```

(3)　保存 9-4.php，然后在上案例中处理添加新增学生信息页面——doInsert.php 中新增 PHP 脚本代码，使之执行完添加操作之后跳转至 9-4.php 页面，显示数据表中的所有学生信息。添加的代码如下。

```
header("location:9-4.php");
```

(4)　保存 doInsert.php 页面。

9.4.4　运行结果

打开浏览器运行 9-3-index.htm 页面，在页面中输入新增学生信息，如图 9-36 所示。



图 9-36　新增学生信息

当输入完毕后，单击【添加】按钮，表单提交至 doInsert.php 页面中进行添加操作，操作完毕后跳转至 9-4.php 页面，查询数据库表中的所有学生信息，并以表格的形式显示在页面中，如图 9-37 所示。



图 9-37　查询并显示所有学生信息

9.4.5　实例分析



源码解析：

在以上案例中的 9-4.php 页面中，首先用 mysql_connect()函数连接 MySQL 数据库，并使用 mysql_select_db()函数选择数据库，然后用 mysql_query()函数执行定义好的用于查询所有学生信息的 SQL 语句，并用 mysql_fetch_array()函数将 mysql_query()返回的结果获取并显示在页面中。mysql_fetch_array()函数返回的是一个数组，以“row[id]”可获取学生表中“id”字段的值，以此类推，可获取学生表中其他字段的值。

9.5　完善学生信息管理系统

在一个成熟的网站系统中，对数据库表的增、删、改、查是不可避免的 4 项操作，在上面一节中已经讲述了对数据库表的查询并显示，本节主要通过示例的方式具体讲解如何在 PHP 程序中插入数据、修改数据和删除数据。

视频教学： 光盘\videos\09\插入数据.avi　　长度：5 分钟
光盘\videos\09\Update 语句.avi　　长度：4 分钟
光盘\videos\09\Delete 语句.avi　　长度：7 分钟

9.5.1　基础知识——插入数据

向数据表中增加一条记录，可以使用下列语句：

```
$mysql_command="insert into <数据表名> ( <字段名 1 >, … , < 字段名 n > ) values ( <
值 1>, … , < 值 n > )";
$result=mysql_query($mysql_command);
```

利用 SQL 命令 insert into 向表中插入新行，数据表名即向其插入数据的表，字段名要与后面的 values 值一一对应。向表中插入数据也可以使用如下语句。

```
$result=mysql_query("insert into <数据表名> ( <字段名 1 >, … , < 字段名 n > ) values
( <值 1> , … , < 值 n > )");
```

在本章的 9.3 节案例中，已经讲述过如何在 PHP 程序中使用 MySQL 函数向数据表中插入数据，这里不再赘述。

9.5.2　基础知识——修改数据

修改数据表中的数据，可以使用如下语句：

```
$mysql_command="update <数据表名> set <字段名 1>= <字段值 1> where <字段名 2><运算
符><字段值 2>";
$result=mysql_query($mysql_command);
```

update 语句是用新值更新现存表中指定的字段值。set 后面“字段名 1”是要修改的列，“字段值 1”是要修改的原始值，where 后面指定要更新的字段名及更新的范围。修改表中的记录还可以使用如下语句：

```
$result=mysql_query("update <数据表名> set <字段名 1>= <字段值 1> where <字段名 2><
运算符><字段值 2>");
```

另外，如何手动将同一数据表内不同字段之间的内容批量转换，可以参考下面的语法：

```
UPDATE table set 字段名=REPLACE(字段名,'原字符串','替换的字符串') where 已知的字段
名 LIKE '%原字符串%'
```

下面创建一个实例，具体讲解如何修改数据，代码如下：

```
<?php
    $link=mysql_connect("localhost:3306","root","000");//连接数据库
    mysql_select_db("user");
    $id="1";                                  //定义要替换的数据编号
    $susername="wanglili";                    //定义要替换的用户名字段值
    $spwd="peng";                             //定义要替换的字符串
    $sql="update tab_user set username='".$susername."' where id=".$id;
                                              //定义修改的 SQL 语句
    $replacesql="UPDATE tab_user set password=REPLACE(password,'yinguopeng',
'mxlygp') where password LIKE '%peng%'";    //定义要替换密码值的 SQL 语句
    if(mysql_query($sql))
```

```
    {
        echo "修改成功";
    }
    else
    {
        echo "修改失败".mysql_error();
    }
    if(mysql_query($replacesql))
    {
        echo "替换成功";
    }
    else
    {
        echo "替换失败".mysql_error();
    }
?>
```

当执行上面一段代码后，查看数据表中的信息：id 为 1 的一条数据中 username 由“maxianglin”变为“wanglili”；password 为“yinguopeng”的一条数据变为了“mxlygp”，并在网页中输出结果如下：

```
修改成功
替换成功
```

9.5.3 基础知识——删除数据

要从数据表中删除记录，可以使用如下语句：

```
$mysql_command="delete from <数据表名> where <字段名> <运算符> <字段值>";
$result=mysql_query($mysql_command);
```

首先通过 delete from 命令删除符合条件的记录，第二行将删除记录的信息赋给变量$result。也可以使用如下格式删除记录：

```
$result=mysql_query("delete from <数据表名> where <字段名> <运算符> <字段值>");
```

在讲解示例之前，先来了解一下 mysql_affected_rows()函数，该函数返回前一次 MySQL 操作所影响的行数，语法格式如下所示：

```
mysql_affected_rows(link_identifier)
```

mysql_affected_rows()函数有一个参数：link_identifier 参数，该参数是可选的，MySQL 的连接标识符。如果没有指定，默认使用最后被 mysql_connect()打开的连接。如果没有找到该连接，函数会尝试调用 mysql_connect()建立连接并使用它。如果发生意外，没有找到连接或无法建立连接，系统将发出 E_WARNING 级别的警告信息。



MySQL_affected_rows()函数用于取得最近一次与 link_identifier 关联的 INSERT，UPDATE 或 DELETE 查询所影响的记录行数。

下面以实例的方式讲解如何删除数据库中的数据，代码如下：

```
<?php
    $link=mysql_connect("localhost:3306","root","000");//连接数据库
    mysql_select_db("user");//选择数据库
    $id=1;//定义要删除的标识列：id 为 1
    $sql="delete from tab_user where id=".$id; //定义删除操作的 SQL 语句
    $result=mysql_query($sql);                 //执行删除操作
    if((mysql_affected_rows()==0)||(mysql_affected_rows()==-1))
                                               //判断受影响的行数
    {
        echo "没有找到记录，或者删除时发生错误".mysql_error();
        exit;
    }
    else {
        echo "该用户信息已经被删除！";
    }
    mysql_close();
?>
```

保存上面的代码并在浏览器中查看运行结果，在页面中输出结果如下：

```
该用户信息已经被删除！
```

这时，数据表中 id 为 1 的记录已经被删除。

9.5.4　实例描述

在前面几节的案例中，已经实现了查询并显示学生信息列表，同时也实现了向数据表中插入一条新增学生信息，而一个学生信息管理系统不可能只简单地增加、查看学生信息，修改、删除学生信息功能也是必不可缺的一部分，那么，下面就来对学生信息管理系统进行完善。

9.5.5　实例应用

【例 9-3】 完善学生信息管理系统

(1)　修改 9-4.php 页面，使学生信息列表中增添一列，代码如下。

```
<?php
    $db_host=localhost;           //MYSQL 服务器名
    $db_user=root;                //MYSQL 用户名
    $db_pass="000";               //MYSQL 用户对应密码
    $db_name="db_student";        //要操作的数据库
    //使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
    $link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器
".mysql_error());
    mysql_select_db($db_name,$link);     //选择相应的数据库，这里选择 test 库
    $sql="select * from student";        //先执行 SQL 语句显示所有记录以与插入后相比较
    $result=mysql_query($sql,$link);     //使用 mysql_query()发送 SQL 请求
```

```
    echo "<table cellSpacing=0 cellPadding=0 width='95%' align=center border=1
style='margin-top:-20px;border-color:#003366'>";     //使用表格格式化数据
    echo "<tr><td style='border-color:#003366'>学生学号</td><td style='border-
color:#003366'>学生姓名</td><td style='border-color:#003366'>学生年龄</td><td
style='border-color:#003366'>学生性别</td><td style='border-color:#003366'>家
庭住址</td><td>操作</td></tr>";
    while($row=mysql_fetch_array($result))  //遍历 SQL 语句执行结果把值赋给数组
    {
        echo "<tr>";
        echo "<td style='border-color:#003366'>".$row[id]."</td>";
        //显示学号
        echo "<td style='border-color:#003366'>".$row[name]." </td>";
        //显示姓名
        echo "<td style='border-color:#003366'>".$row[age]." </td>";
        //显示年龄
        echo "<td style='border-color:#003366'>".$row[sex]." </td>";
        //显示性别
        echo "<td style='border-color:#003366'>".$row[address]." </td>";
        //显示地址
        echo "<td style='border-color:#003366'><a href='9-5.php?id=".$row[id]."&sign=
update'>修改</a>  <a href='9-5.php?id=".$row[id]."&sign=del'>删除
</a></td>";
        echo "</tr>";
    }
    echo "<tr>";
    echo "<td colspan=6 align='right'><a href='9-3.php' style='font-size:14px;
font-weight:bold;'>添加学生信息</a></td>";
    echo "</tr>";
    echo "</table>";
?>
```

当用户单击记录行后面的【修改】超链接或者【删除】超链接时，跳转至 9-5.php 页面。在单击超链接时，传递了两个参数：一个是要操作的记录 id，另一个是要操作的类型。如果是修改记录数据，则 sign 的值为 update，如果是删除，则为 del。

(2) 新建 PHP 文件，命名为 9-5.php。

(3) 编辑 9-5.php，判断当前用户要执行的操作。如果当前用户要删除某条记录，则执行删除操作即可；如果当前用户要修改某条记录，则需要从数据表中查询该条记录信息，并跳转至 doUpdate.php 页面，在该页面中显示记录信息。9-5.php 中的 PHP 脚本代码如下。

```
<?php
    $db_host=localhost;          //MYSQL 服务器名
    $db_user=root;               //MYSQL 用户名
    $db_pass="000";              //MYSQL 用户对应密码
    $db_name="db_student";       //要操作的数据库
    //使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
    $link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器
".mysql_error());
    mysql_select_db($db_name,$link);    //选择相应的数据库，这里选择 db_student 库
```

```
    $sign=$_GET['sign'];      //获取操作标识
    $id=$_GET['id'];          //获取操作 id
    $name="";                 //存储学生姓名
    $age=0;                   //存储学生年龄
    $sex="";                  //存储学生性别
    $address="";              //存储学生住址
    if($sign=="update")
    {
        $sql="select * from student where id=".$id;
        $result=mysql_query($sql);
        //获取查询结果中指定字段的值
        $name=mysql_result($result,0,"name");
        $age=mysql_result($result,0,"age");
        $sex=mysql_result($result,0,"sex");
        $address=mysql_result($result,0,"address");
        mysql_close();

    header("location:doUpdate.php?id=".$id."&name=".$name."&age=".$age."&se
x=".$sex."&address=".$address);
    }
    else
    {
        $sql="delete from student where id=".$id;  //删除语句
        $result=mysql_query($sql);                 //执行 SQL 语句
        if($result)
        {
            echo "学号为：".$id."的数据已经删除！<br/><a href='9-4.php'>返回学生信
息列表</a>";
        }
        else
        {
            echo "抱歉，删除失败！";
        }
        mysql_close();
    }
?>
```

当用户需要修改某条记录时，查询该条记录的所有信息，并将获取到的所有信息作为参数传递给 doUpdate.php 页面。

(4) 新建 doUpdate.php 页面，在该页面中获取 9-5.php 页面传递过来的参数，并显示在表单元素中，代码如下。

```
<?php
    $id=$_GET['id'];              //获取 ID
    $name=$_GET['name'];          //获取姓名
    $age=$_GET['age'];            //获取年龄
    $sex=$_GET['sex'];            //获取性别
    $address=$_GET['address'];    //获取地址
?>
```

```
<form name="myForm" method="post" action="update.php">
    <input type="hidden" name="id" value="<?php echo $id;?>" />
    <table cellpadding="0" cellspacing="0" border="1" width="95%" style=
"margin-top:-50px;">
        <tr>
            <td>学号</td>
            <td><input type="text" name="txtId" value="<?php echo $id;?>"/></td>
        </tr>
        <tr>
            <td>姓名</td>
            <td><input type="text" name="txtName" value="<?php echo $name;?>"/></td>
        </tr>
        <tr>
            <td>年龄</td>
            <td><input type="text" name="txtAge" value="<?php echo $age;?>"></td>
        </tr>
        <tr>
            <td>性别</td>
            <td>
                <select name="sSex">
                    <option value="male"
                    <?php
                        if($sex=="male")
                        {
                            echo "selected=selected";
                        }
                    ?>
                    >male</option>
                    <option value="female"
                    <?php
                            if($sex=="female")
                            {
                                echo "selected=selected";
                            }
                    ?>
                    >female</option>
                </select>
            </td>
        </tr>
        <tr>
            <td>地址</td>
            <td><input type="text" name="txtAddress" value="<?php echo
            $address;?>"/></td>
        </tr>
        <tr>
            <td colspan="2" align="center">
                <input type="submit" name="submit" value="修改"/>
                <input type="reset" name="reset" value="重置">
            </td>
```

```
        </tr>
    </table>
</form>
```

(5)　编辑表单提交页面 update.php，处理表单数据，修改数据表信息，代码如下：

```
<?php
    $link=mysql_connect("localhost:3306","root","000");
    mysql_select_db("db_student");
    $id=$_POST['id'];//获取要修改的数据编号
    $stunum=$_POST['txtId']|null;
    $name=$_POST['txtName'];
    $age=$_POST['txtAge'];
    $sex=$_POST['sSex'];
    $address=$_POST['txtAddress'];
    $exec="update student set id=".$stunum.",name='".$name."',age=".$age.",
sex='".$sex."',address='".$address."' where id=".$id;  //修改语句
    echo $exec;
    $result=mysql_query($exec);
    if($result) echo "学号为:".$id."的学生信息已经修改成功!<br/><a href='9-4.php'>
返回学生信息列表页面</a>";
    else echo "抱歉，修改失败！<br>";
    mysql_close();
?>
```

(6)　新建 HTML 文件，命名为 9-5-index.htm。使用 Frameset 将 header.htm、menu.htm 和 9-4.php 页面整合，分别作为 Frameset 的顶部页面、左边菜单导航页面和主页面。

9.5.6　运行结果

打开 IE 浏览器，运行 9-5-index.htm 页面，出现如图 9-38 所示的页面效果。



图 9-38　学生信息列表页面效果

单击学生信息列表中学生学号为“1001”记录后面的【修改】超链接，页面跳转至 9-5.php 页面，查询该条记录的信息，然后再次跳转至 doUpdate.php 页面，将查询的结果显示在表单域中，如图 9-39 所示。



图 9-39　修改输入页面效果

当用户将学生姓名由“马向林”修改为“王丽丽”后，单击【修改】按钮，表单提交至 update.php 页面，进行修改操作。如果修改成功，则在 update.php 页面中输出“学号为：1001 的学生信息已经修改成功！”，并在下面有一个【返回学生信息列表页面】的超链接，单击该超链接，跳转至 9-4.php 页面，将修改后的数据显示在列表中。

当用户单击学生信息列表页面中的【删除】超链接时，连接至 9-5.php 页面，在该页面中执行删除操作。如果删除成功，在该页面中提示成功信息，如图 9-40 所示。



图 9-40　删除成功页面效果

9.5.7　实例分析


源码解析：

上面实例完成了学生信息管理系统中的删、改、查 3 项操作。当用户需要修改某条记录的数据时，单击学生信息列表中的【修改】超链接，将学生学号传递到 9-5.php 页面中，在该页面中判断 sign 的值，如果是“update”，表明用户单击的是【修改】超链接，在该页面中查询用户要修改的记录信息，并使用 header()函数跳转至 doUpdate.php 页面。在 doUpdate.php 页面中获取用户要修改的记录信息，并显示在表单域中。当用户输入修改后的学生信息后，单击 doUpdate.php 页面中的【修改】按钮，表单提交到 update.php 页面，在该页面中执行修改操作。

9.6　获取无法连接数据库的错误信息

通过前 5 节的学习，掌握了如何连接数据库和对数据库数据的基本操作，那么，当执行 SQL 语句出现错误时，如何获取错误信息呢？本节将为大家讲解在执行 SQL 语句时出现错误，获取错误信息的一些相关函数。


视频教学：光盘\videos\09\获取错误信息.avi　　长度：7 分钟

9.6.1　基础知识——获取错误信息

在执行 SQL 语句时，难免会出错，出错之后应该如何解决问题呢？首先需要获得错误信息，然后再根据错误信息修改错误。在 PHP 中有两个获取错误的内置函数，分别是 mysql_error()函数和 mysql_errno()函数，本节将针对这两个函数进行详细的介绍。

1．mysql_error()函数

mysql_error()函数返回上一个 MySQL 操作产生的文本错误信息，如果 MySQL 操作没有出错，则返回空字符串，该函数的语法格式如下：

```
mysql_error(connection)
```

mysql_error()函数有一个可选参数，规定 SQL 连接标识符，如果未规定，则使用上一个成功打开的连接从 MySQL 服务器中提取错误信息。mysql_error()函数仅返回最后一次 MySQL 函数执行的错误文本，不包括 mysql_error()函数和 mysql_errno()函数因此如果要使用此函数，应确保在调用另一个 MySQL 函数之前检查其值。下面创建一个使用该函数的实例，具体代码如下：

```
<?php
    $con = mysql_connect("localhost","wrong_user","wrong_pwd");//连接数据库，使用错误的用户名和错误的密码
    if (!$con)
```

```
    {
        die(mysql_error());
    }
    mysql_close($con);
?>
```

保存并执行该段代码，结果如下：

```
Access denied for user 'wrong_user'@'localhost' (using password: YES)
```

2. mysql_errno()函数

mysql_errno()函数和 mysql_error()函数功能相同，只是 mysql_errno()函数返回的是错误信息的数字编码，并不返回错误的详细信息，如果执行过程没有出错，则返回 0，该函数的语法格式如下：

```
mysql_errno(connection)
```

下面创建一个实例，具体讲解如何运用 mysql_errno()函数。该实例使用错误的用户名和密码连接数据库，从而返回错误信息，具体代码如下：

```
<?php
    $con = mysql_connect("localhost","wrong_user","wrong_pwd");//连接数据库，使
用错误的用户名和错误的密码
    if (!$con)
    {
        echo "不能连接数据库的错误信息编号：".mysql_errno()."<br/>";
        echo "不能连接数据库的错误信息为：".mysql_error();
    }
    mysql_close($con);
?>
```

保存并执行该段代码，在页面中输出结果如下所示：

```
不能连接数据库的错误信息编号：1045
    不能连接数据库的错误信息为：Access denied for user 'wrong_user'@'localhost'
(using password: YES)
```

9.6.2 实例描述

当用户阅读一条新闻时，需要从数据表中查询该条新闻信息并显示在页面中，这时就需要打开可用的链接，与数据库进行交互。而当用户名或者密码错误时，无法使 PHP 与数据库连接，从而无法正常显示新闻动态，这时就需要在页面中提示用户错误信息。下面就来实现新闻信息读取失败提示错误信息这样的一个功能。

9.6.3 实例应用

【例 9-4】 获取无法连接数据库的错误信息

(1) 新建 PHP 文件，命名为 9-6.php。

(2) 在页面中编辑 PHP 代码，使用错误的用户名和正确的密码与数据库进行连接，并查询新闻动态数据。该案例使用的数据库为 user 数据库，表为 tab_content。代码如下。

```
<?php
        $db_host=localhost;           //MYSQL 服务器名
        $db_user="wrong_user";        //MYSQL 用户名
        $db_pass="000";               //MYSQL 用户对应密码
        $db_name="user";              //要操作的数据库
        //使用 mysql_connect()函数对服务器进行连接，如果出错返回相应信息
        $link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error());
        mysql_select_db($db_name,$link);    //选择相应的数据库，这里选择 user 库
        $sql="select * from tab_content";   //先执行 SQL 语句显示所有记录以与插入后相比较
        $result=mysql_query($sql,$link);    //使用 mysql_query()发送 SQL 请求
        //获取查询结果中指定字段的值
        $name=mysql_result($result,0,"name");
        $content=mysql_result($result,0,"content");
        echo "作者:".$name."<br/>";
        echo $content;
        mysql_close();
?>
```

(3) 保存 9-6.php。

9.6.4 运行结果

在浏览器中运行 9-6.php 页面，数据库连接失败，提示错误信息，如图 9-41 所示。



图 9-41 数据库连接失败提示错误信息

9.6.5 实例分析



源码解析：

在上面的实例中，使用了“$link=mysql_connect($db_host,$db_user,$db_pass)or die("不能连接到服务器".mysql_error())”语句来获取与数据库的连接，而在上面定义的$db_user 变量所表示的用户名是错误的，因此在与数据库连接时会发生错误，从而执行“die("不能连接到服务器".mysql_error())”，在页面中提示错误信息，否则显示数据表中的新闻信息。

9.7 制作显示服务器中的数据库和对应的表信息

在前面的章节中，讲解了如何使用 PHP 内置函数获得数据库中的数据。同样，在 PHP 中使用内置函数也可以获得数据库和表相关信息。本节将为大家讲解如何通过 PHP 内置函数获取数据库和表信息。

**视频教学：**光盘\videos\09\获取数据库和表信息.avi　　长度：8 分钟

9.7.1 基础知识——获取数据库信息

在进行数据库操作时，有时可能因程序的需求需要获取数据库的相关信息如数据库列表等信息。本节将对在 PHP 中如何获取 MySQL 服务器的相关信息进行详细讲解。

1．获取数据库列表

获取 MySQL 服务器的数据库列表信息，可使用 mysql_list_dbs()函数。该函数的语法格式如下：

```
resource mysql_list_dbs ( [resource link_identifier] )
```

其中，参数 link_identifier 为数据库连接标识符。若未设置该标识符，函数将使用上一个已经打开的数据库连接；若没有可用的数据库连接，函数将尝试以无参方式调用 mysql_connect()函数建立数据库连接。函数返回一个结果指针，包含当前 MySQL 进程中所有可用的数据库，可使用 mysql_fetch_array()类似的函数获取其结果集。下面创建一个实例，具体讲解如何运用该函数，代码如下：

```
<?php
    $con = mysql_connect("localhost", "root", "000") or die("数据库连接失败");
    //连接数据库信息
    $dbs=mysql_list_dbs();
    //获取数据库列表
    while ($dbrow=mysql_fetch_array($dbs,MYSQL_BOTH)) {
    //获取结果集
        echo $dbrow[0]."<br/>";
```

```
    }
    mysql_close($con);
?>
```

保存并执行该段代码，在页面中输出服务器中所有的数据库名称，结果如下：

```
information_schema
db_student
mysql
test
user
userdb
```

在上述程序中，首先连接数据库服务器，再使用 mysql_list_dbs()函数获取服务器数据库列表，最后使用 mysql_fetch_array()函数获取结果集，并将结果集显示出来。不难发现，使用该方式与在 MySQL 命令行使用“show databases”命令的结果是相同的。

2．获取数据库名称

mysql_db_name()函数取得 mysql_list_dbs()调用所返回的数据库名。该函数的语法格式如下：

```
mysql_db_name(list, row, field)
```

mysql_db_name()函数有 3 个参数。

- List：该参数是必需的，表示 mysql_list_dbs()调用所返回的结果指针。
- row：该参数是必需的，表示结果集中的行号，从 0 开始。
- field：该参数是可选的，表示结果集中的字段名。

如果执行成功，则返回数据库名，失败则返回 false。如果返回了 false，可以使用 mysql_error()来判断错误的种类。下面创建一个实例，具体讲解该函数的使用，代码如下：

```
<?php
    $con = mysql_connect("localhost", "root", "000");  //获取数据库连接
    if (!$con)
    {
        die('Could not connect: ' . mysql_error());
    }
    $db_list = mysql_list_dbs($con);//获取服务器中所有的数据库
    $i = 0;
    $db_count = mysql_num_rows($db_list);
                                //使用 mysql_num_rows()函数获取数据库的个数
    while ($i < $db_count)      //循环遍历服务器中所有的数据库
    {
        echo mysql_db_name($db_list, $i) . "<br />";   //逐行输出数据库名称
        $i++;
    }
    mysql_close($con);
?>
```

保存并执行该段代码，在页面中输出服务器中所有数据库的名称，与使用 mysql_list_dbs()函数输出服务器中数据库名称的结果一样。

9.7.2 基础知识——获取数据表信息

对于显示某个数据库中的数据表的信息，首先需要选择数据库，再使用 mysql_list_tables()函数获取数据表列表。其语法格式如下所示：

```
resource mysql_list_tables ( string database [, resource link_identifier] )
```

其中，参数 database 为要获取数据表列表的数据库名，参数 link_identifier 为数据库连接标识符。若未设置该参数，函数将使用上一个已打开的数据库连接。若没有可用的数据库连接，函数将尝试以无参数方式调用 mysql_connect()函数创建数据库连接；函数返回一个包含指定数据库的数据表列表的结果集，可使用类似 mysql_fetch_array()函数获取该结果集，若失败则返回 false。下面创建一个实例，讲解如何使用 mysql_list_tables()函数获取指定数据库中的数据表列表，代码如下：

```
<?php
    mysql_connect("localhost","root","000") or die("数据库连接失败！");
    //连接数据库服务器
    $tables=mysql_list_tables("db_student");
    //获取数据库 db_student 的数据表列表
    if($tables)
    {
        while ($tabrow=mysql_fetch_array($tables)) //获取结果集
        {
            echo $tabrow[0]."<br/>";

        }
    }
    else
    {
        echo "获取数据表失败！";
    }
    mysql_close();
?>
```

在上述代码中，首先连接数据库服务器，再使用 mysql_list_tables()函数获取指定数据库中的数据表列表信息。

9.7.3 实例描述

在一个企业的内部管理系统中，有时需要在页面中显示服务器中所有的数据库和表信息，

以供操作员对其进行操作。下面就来制作显示数据库和表信息的后台页面。

9.7.4　实例应用

【例 9-5】 制作显示服务器中的数据库和对应的表信息

(1)　新建 PHP 文件，命名为 9-7.php。

(2)　编辑 9-7.php，在页面中使用 mysql_list_dbs()函数获取服务器中所有数据库信息，并使用 mysql_tablename()获取当前数据库中的列表信息，最后将获取到的数据库和表信息显示在页面中，代码如下：

```
<?php
    //连接数据库
   $connection=mysql_connect("localhost","root","000");
   //获取数据库列表
   $result_db=mysql_list_dbs($connection);
   //获取数据库数量
   $result_count=mysql_num_rows($result_db);
   for($i=0;$i<$result_count;$i++)
  {
        //获取数据库名称
        $db_name=mysql_tablename($result_db,$i);
        echo "数据库为：".$db_name."的表如下：<br/>";
        //获取当前数据库中的数据列表
        $result_table=mysql_list_tables($db_name,$connection);
        //获取数据库数据列表数量
        $result_tabcount=mysql_num_rows($result_table);
        for($j=0;$j <$result_tabcount;$j++)
        {
            $table_name=mysql_tablename($result_table,$j);
            echo "      表".($j+1).":".$table_
name."]<br/>";

       }
   }
?>
```

(3)　编辑 top.html 和 left.html 页面，分别作为框架页面的顶部页面和左边菜单导航页面。

(4)　新建 HTML 文件，命名为 9-7-index.html。

(5)　在 9-7-index.html 页面中使用 Frameset 将 top.html、left.html 和 9-7.php 页面整合。

9.7.5　运行结果

在浏览器中运行 9-7-index.html 页面，在页面中输出服务器中所有的数据库和对应的表列表信息，如图 9-42 所示。

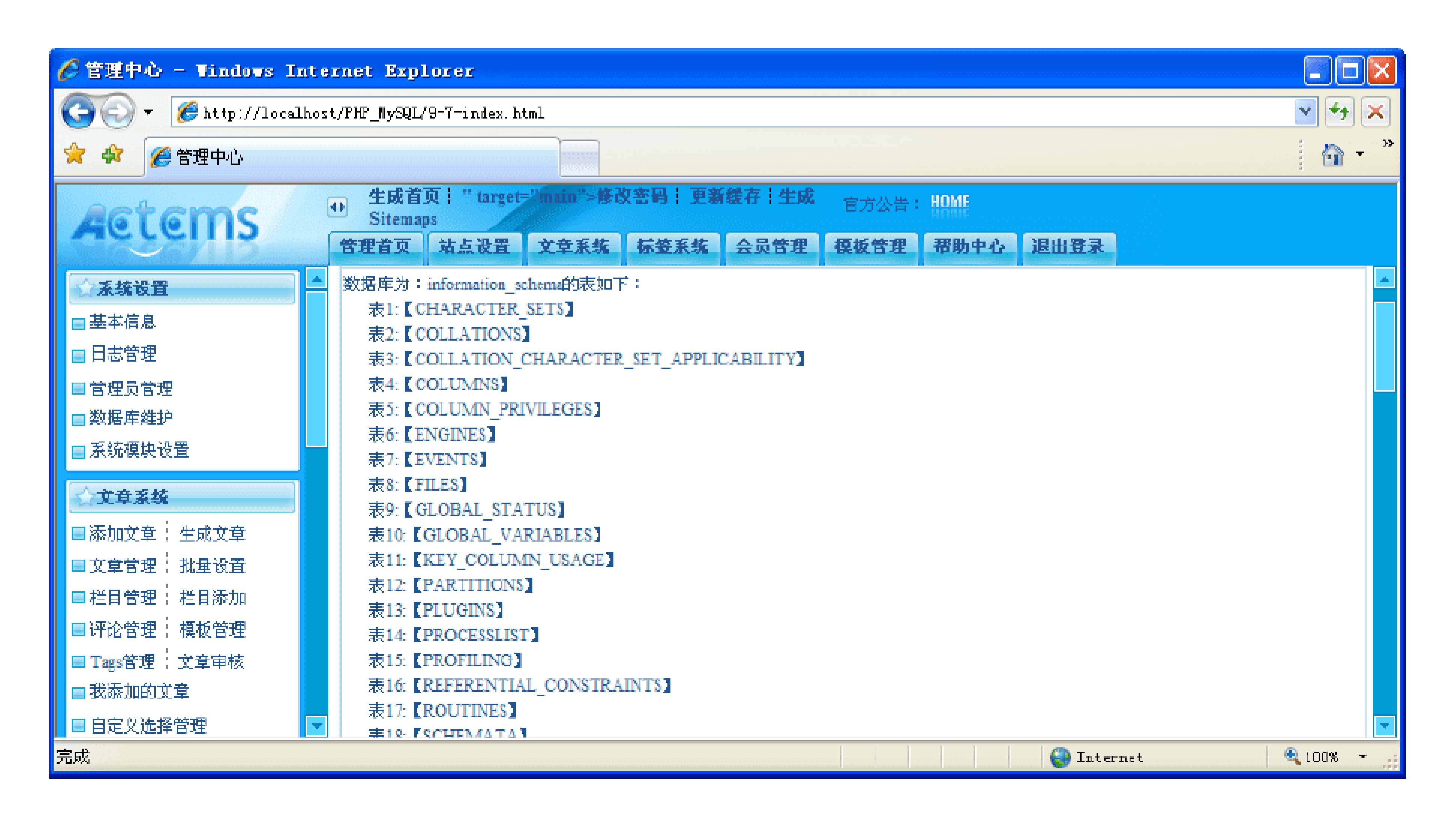


图 9-42　获取数据库和对应的表信息

9.7.6　实例分析



源码解析：

在上面实例中的 9-7.php 页面中，首先使用 mysql_list_dbs()函数获取服务器中的所有数据库列表信息，接着循环遍历数据库列表信息，使用 mysql_tablename()函数获取数据库的名称信息。然后使用 mysql_list_tables()函数获取该数据库中的所有表信息，接着循环遍历该数据库表信息，使用 mysql_tablename()函数获取指定数据库中的数据表信息。

9.8　实现显示某张数据表中所有字段信息的功能

在应用中，有时需要获取数据表的列的信息，如数据表的列的数目、名称、长度等。在 PHP 中，有一组内置函数专门用于获取 MySQL 服务器数据库中表字段的相关信息。本节主要介绍获取字段信息的内置函数。



视频教学：光盘\videos\09\获取字段信息.avi　　长度：13 分钟

9.8.1　基础知识——获取字段信息

获取数据表的列的信息在实际开发中是常用的，下面具体讲述如何运用 PHP 中的内置函数获取数据表中的字段信息。

1．获取列的所有信息

获取列的所有信息，包括列名称、表名称、默认值、长度等，可使用 mysql_fetch_field()

函数，该函数从结果集中取得列信息并作为对象返回，该函数的语法格式如下：

```
object mysql_fetch_field ( resource result [, int field_offset] )
```

mysql_fetch_field()函数可以用来从某个查询结果中取得字段的信息，该函数有两个参数。result 参数是必需的，表示要使用的数据指针，该数据指针是从 mysql_query()函数中返回的结果；field_offset 参数是可选的，表示字段偏移量，规定从哪个字段开始，0 指第一个字段，如果没有设置该参数，则从下一个尚未被 mysql_fetch_field()函数取得的字段被提取。该函数返回的对象属性为字段的相关信息，例如列名和该列所在表的表名，这些属性如下。

- name：列名。
- table：该列所在的表名。
- max_length：该列最大长度。
- not_null：如果该列不能为 NULL，返回 1，否则返回 0。
- primary_key：如果该列是主键，返回 1，否则返回 0。
- unique_key：如果该列是 unique key，返回 1，否则返回 0。
- multiple_key：如果该列是 non-unique key，返回 1，否则返回 0。
- numeric：如果该列是 numeric，返回 1，否则返回 0。
- blob：如果该列是 BLOB，返回 1，否则返回 0。
- type：该列的类型。
- unsigned：如果该列是无符号数，返回 1，否则返回 0。
- zerofill：如果该列是 zero-filled，返回 1，否则返回 0。

下面创建一个示例，具体讲解如何运用该函数，代码如下：

```
<?php
    $con = mysql_connect("localhost", "root", "000") die('Could not connect:
        ' . mysql_error());                                          //获取与数据库连接
    $db_selected = mysql_select_db("user",$con);
    $sql = "SELECT * from tab_user";
    $result = mysql_query($sql,$con);
    echo "<table><tr>";
    while ($property = mysql_fetch_field($result)) //获取表中的字段信息
    {
        echo "<td width='300px'>";
        echo "Field name: " . $property->name . "<br />";
        echo "Table name: " . $property->table . "<br />";
        echo "Default value: " . $property->def . "<br />";
        echo "Max length: " . $property->max_length . "<br />";
        echo "Not NULL: " . $property->not_null . "<br />";
        echo "Primary Key: " . $property->primary_key . "<br />";
        echo "Unique Key: " . $property->unique_key . "<br />";
        echo "Mutliple Key: " . $property->multiple_key . "<br />";
        echo "Numeric Field: " . $property->numeric . "<br />";
        echo "BLOB: " . $property->blob . "<br />";
        echo "Field Type: " . $property->type . "<br />";
        echo "Unsigned: " . $property->unsigned . "<br />";
        echo "Zero-filled: " . $property->zerofill;
```

```
        echo "</td>";
    }
    echo "</tr></table>";
    mysql_close($con);
?>
```

在上面的代码中，运用 mysql_fetch_field()函数返回的对象，获取了字段的相关属性。保存并执行该段代码，页面效果如图 9-43 所示。

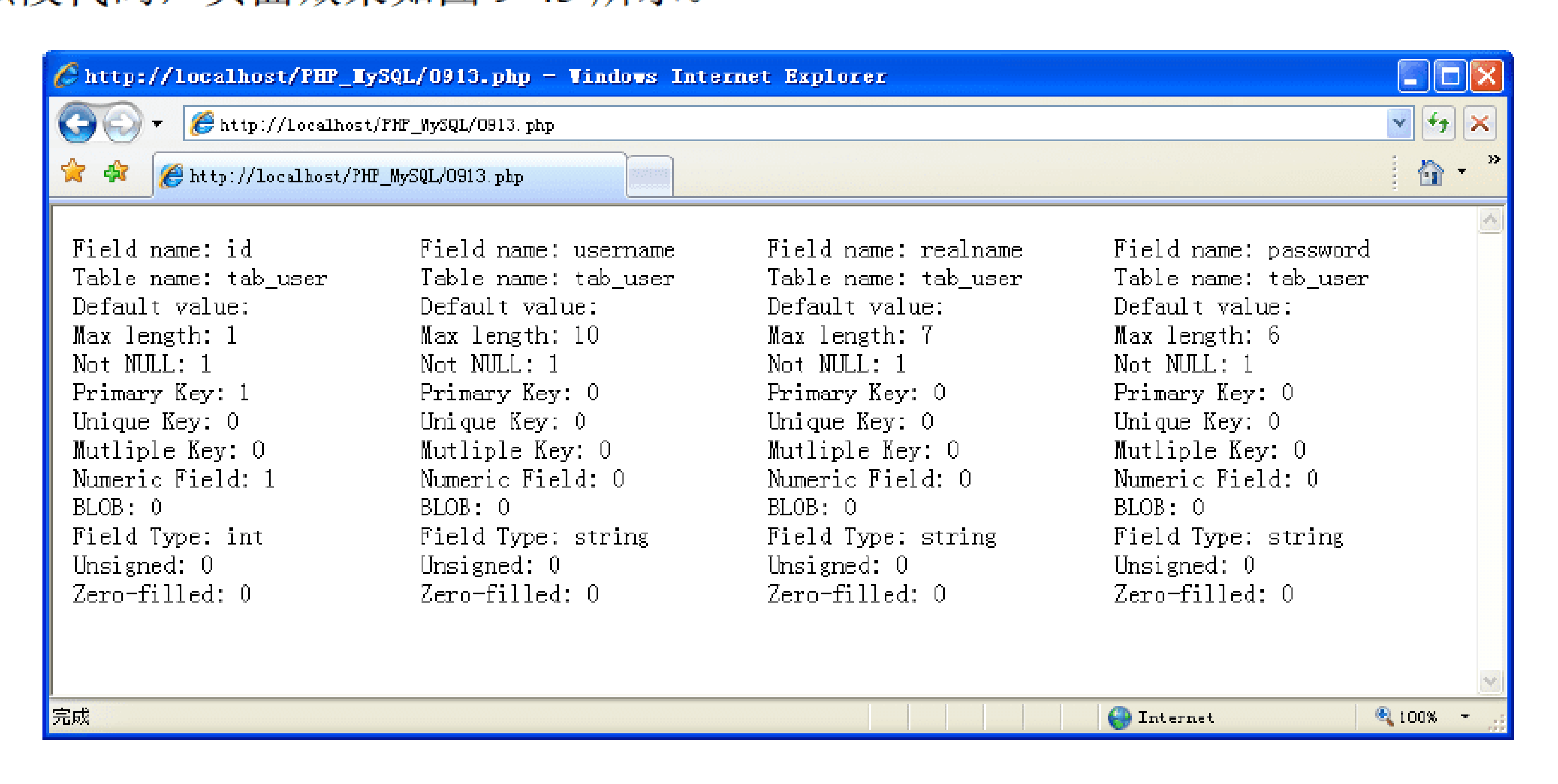


图 9-43　运用 mysql_fetch_field()函数页面效果图

2．获取列的数目

获取数据表的列的数目，可使用 mysql_num_fields()函数。其语法格式如下所示：

```
int mysql_num_fields ( resource result )
```

其中，参数 result 为使用 mysql_query()函数所返回的结果集。函数返回结果集中列的数目。



函数返回的是结果集中列的数目。因此，若需要获取数据表的列的数目，在使用 mysql_query()函数执行 SQL 语句时，使用的查询语句应使用“*”代表所有的列。

下面创建一个实例，具体讲解如何使用该函数获得字段的数量，代码如下：

```
<?php
    $con = mysql_connect("localhost", "root", "000");  //获取与数据库的连接
    if (!$con)
    {
      die('Could not connect: ' . mysql_error());
    }
    $db_selected = mysql_select_db("user",$con);       //选择数据库
    $sql = "SELECT * FROM tab_user";
    $result = mysql_query($sql,$con);                  //执行查询操作
    echo mysql_num_fields($result);                    //输出表中字段数量
    mysql_close($con);
?>
```

保存并执行该段代码，结果如下：

```
4
```

在上述的示例中，首先连接数据库服务器，然后使用 mysql_query()函数执行一个包含所有列的 SQL 语句，再根据返回的结果集使用 mysql_num_fields()函数获取结果集中的列的数目。

3．获取指定字段关联的标志

从结果集中取得和指定字段关联的标志，比如 primary-key、enum 和 binary 等，可使用 mysql_field_flags()函数，其语法格式如下：

```
string mysql_field_flags ( resource result, int field_offset )
```

mysql_field_flags()函数返回指定字段的字段标志。每个标志都有一个单词表示，之间用一个空格分开，因此可使用 explode()将其分开。该函数有两个参数：result 参数表示要使用的数据指针，该数据指针是从 mysql_query()函数返回的结果；field_offset 表示从哪个字段开始返回，0 指示第一个字段。如果 MySQL 版本足够新，则会支持以下的标志：

- auto_intcrement
- binary
- blob
- enum
- multiple_key
- not_null
- primary_key
- timestamp
- unique_key
- unsigned
- zerofill

下面创建一个示例，具体讲解如何运用 mysql_field_flags()函数，代码如下：

```
<?php
    $con = mysql_connect("localhost", "root", "000");  //获取与数据库的连接
    if (!$con)
    {
      die('Could not connect: ' . mysql_error());
    }
    $db_selected = mysql_select_db("user",$con);       //选择数据库
    $sql = "SELECT * from tab_user";
    $result = mysql_query($sql,$con);                  //执行查询操作
    $flags = mysql_field_flags($result, 0);            //获取给定字段关联的标志
    echo $flags;                                       //输出字段标志
    mysql_close($con);
?>
```

保存并执行该段代码，结果如下：

```
not_null primary_key auto_increment
```

4. 获取列的长度

获取数据表中列的长度，可使用 mysql_field_len()函数，其语法格式如下：

```
int mysql_field_len ( resource result, int field_offset )
```

其中，参数 result 为使用 mysql_query()函数执行 SQL 语句所返回的结果集，参数 filed_offset 为列的偏移量。函数返回结果集中指定列的长度。

下面创建一个示例，来具体讲解如何使用 mysql_field_len()函数获取数据表中指定字段的长度，代码如下：

```
<?php
    $con = mysql_connect("localhost", "root", "000");  //获取与数据库的连接
    if (!$con)
    {
      die('Could not connect: ' . mysql_error());
    }
    $db_selected = mysql_select_db("user",$con);       //选择数据库
    $sql = "SELECT * from tab_user";
    $result = mysql_query($sql,$con);                  //执行查询操作
    $length = mysql_field_len($result, 0);             //获取第一个字段的长度
    echo $length;                                      //输出长度
    mysql_close($con);
?>
```

保存并执行该段代码，在页面中输出第一个字段的长度，结果为如下。

```
11
```

在上述示例中，首先获取返回结果集中列的总数，再使用 mysql_field_len()函数获取数据表 tab_user 表中第一列的长度。

5. 获取列的名称

获取数据表的列的名称，其实在前面使用 mysql_fetch_array()函数获取结果集时，若采用参数 MYSQL_ASSOC 或 MYSQL_BOTH 获取的结果集数组中键名就是列的名称。除此之外，PHP 还提供了一个专门获取列名称的函数 mysql_field_name()，其语法格式如下：

```
string mysql_field_name ( resource result, int field_offset )
```

其中，参数 result 表示要使用的数据指针，该数据指针是从 mysql_query()函数返回的结果；field_offset 参数指定从哪个字段开始返回，0 指示第一个字段。此函数的使用与上面所讲解的 mysql_field_len()函数的使用相同，这里不再赘述。

6. 获取列的类型

获取数据表列的类型，可使用 mysql_field_type()函数。该函数如果执行成功，则返回指定字段的类型，如果失败，则返回 false。其语法格式如下：

```
string mysql_field_type ( resource result, int field_offset )
```

参数含义与 mysql_field_name()函数的参数含义相同。

7．获取指定字段所在的表名

获取结果集中指定字段所在的表名，可使用 mysql_field_table()函数，如果失败，则返回 false。其语法格式如下：

```
string mysql_field_table ( resource result, int field_offset )
```

参数含义与 mysql_field_name()函数含义相同。

9.8.2　实例描述

在一个企业的内部管理系统中有时需要存在一个可以读取某张数据表中的字段信息页面，包括字段类型、字段名称、字段长度和其他相关信息。下面就来做一个可以显示某张数据表中字段信息的页面。

9.8.3　实例应用

【例 9-6】 实现显示某张数据表中所有字段信息的功能

(1)　新建 PHP 文件，命名为 9-8.php。

(2)　编辑 9-8.php，在页面中获取与数据库的连接，并指定要获取的数据表，从而使用 PHP 函数获取数据表中所有字段信息，代码如下：

```
<?php
    $hostname="localhost";      //定义连接到的 mysql 服务器名
    $username="root";           //定义用于连接的用户名
    $password="000";            //定义用于连接的密码
    $link=mysql_connect($hostname,$username,$password);    //打开 mysql 连接
    if(!$link)
    {
        die('连接数据库失败的原因是：'.mysql_error());
    }
    mysql_select_db('db_student');                     //选择数据库
    $sql_str="select * from student where id=1";    //定义 SQL 语句
    $result=mysql_query($sql_str);                     //执行 SQL 语句
    $re_len=mysql_num_fields($result);                 //获取结果集中字段的数量
    echo "<table width='95%'><tr>";
    for($i=0;$i<$re_len-1;$i++)
    {
        echo "<td>";
        echo "名称为【".mysql_field_name($result,$i)."】的列信息如下：<br/>";
        echo "      长度为   
            ".mysql_field_len($result,$i)."<br/>";
        echo "      类型为   
            ".mysql_field_type($result,$i)."<br/>";
        echo "      详细信息:<br/>";
        $meta=mysql_fetch_field($result);       //获取字段信息
```

```
        if(!$meta)                                    //如果值不存在
        {
            echo "no information available<br/>n";  //输出无可用信息
        }
        echo "<pre>
    blob:     $meta->blob
    max_length:   $meta->max_length
    multiple_key:  $meta->multiple_key
    name:       $meta->name
    not_null:      $meta->not_null
    numeric:      $meta->numeric
    primary_key:  $meta->primary_key
    table:         $meta->table
    type:         $meta->type
    unique_key:  $meta->unique_key
    unsigned:    $meta->unsigned
    zerofill:      $meta->zerofill
            </pre>";                                  //结束去格式输出
        echo "</td>";
    }
    echo "</table>";
    mysql_close($link);
?>
```

(3) 新建 HTML 文件，命名为 9-8-index.htm。

(4) 在 9-8-index.htm 页面中使用 Frameset 将前面实例中使用到的 header.htm、menu.htm 和 9-8.php 页面整合，分别作为 Frameset 的顶部页面、左边菜单导航页面和主页面。

9.8.4 运行结果

在浏览器中运行 9-8-index.htm，在页面中输出表中字段的详细信息，如图 9-44 所示。



图 9-44　学生表字段信息

9.8.5　实例分析



源码解析：

在上面实例中的 9-8.php 页面中，获取与数据库的连接并执行查询 id 为 1 的记录后，首先使用 mysql_num_fields()函数获取数据表字段数量，接着循环遍历(获取到的所有字段个数-1)个字段信息，并将每个字段的名称、类型、长度和详细信息输出到页面中。

9.9　实现查看服务器和客户端库的信息功能

PHP 提供了许多内置函数用于获取客户端或 MySQL 服务器端信息。本节将介绍 PHP 中的一些辅助函数。



视频教学：光盘\videos\09\辅助函数.avi　　　　长度：4 分钟

9.9.1　基础知识——辅助函数

下面简单介绍 PHP 中的一些辅助函数，这些函数使得 PHP 可以轻松地获取客户端或 MySQL 服务器端的信息。

1．获取当前连接的字符编码集

获取当前连接的字符编码集，可使用 mysql_client_encoding()函数。其语法格式如下：

```
string mysql_client_encoding ( [resource link_identifier] )
```

该函数有一个 link_identifier 参数是必需的，表示 MySQL 的连接标识符，如果没有指定，默认使用最后将 mysql_connect()打开的连接。如果没有找到该连接，函数会尝试调用 mysql_connect()建立连接并使用它，如果发生意外，没有找到连接或无法建立连接，系统发出 E_WARNING 级别的警告信息。

2．获取 MySQL 服务器信息

获取 MySQL 服务器信息，可使用 mysql_get_server_info()函数，如果执行成功，则返回 MySQL 服务器的版本号，否则返回 false。其语法格式如下：

```
string mysql_get_server_info ( [resource link_identifier] )
```

参数 link_identifier 为可选参数，表示 MySQL 连接，如果未规定，则使用上一个连接。

3．获取 MySQL 主机信息

获取 MySQL 主机信息，可使用 mysql_get_host_info()函数，如果成功，则返回当前正在使用的 MySQL 连接的类型，如果失败，则返回 false。其语法格式如下：

```
string mysql_get_host_info ( [resource link_identifier] )
```

4．获取 MySQL 客户端信息

获取 MySQL 客户端信息，可使用 mysql_get_client_info()函数，如果成功，则返回 MySQL 客户端的版本号，否则返回 false。其语法格式如下：

```
string mysql_get_client_info ( void )
```

5．获取服务器状态

获取服务器状态，可使用 mysql_stat()函数。该函数返回 MySQL 服务器的当前系统状态，例如正常运行时间、线程、执行的查询、已经打开的表、已经刷新的表、当前打开的表和每秒平均查询数等。如果要获得其他状态变量的完整列表，只能使用 SQL 命令 SHOW STATUS，其语法格式如下：

```
string mysql_stat ( [resource link_identifier] )
```

9.9.2 实例描述

在网站的后台管理系统中可以查看本系统所使用的数据库、服务器和客户端的一些信息是多么美妙的一件事，那么，下面就来实现这样的功能吧！

9.9.3 实例应用

【例 9-7】 实现查看服务器和客户端库的信息功能

(1) 新建 PHP 文件，命名为 9-9.php。

(2) 编辑 9-9.php，在页面中使用 PHP 的一些函数获取 MySQL 服务器和客户端的相关信息，代码如下：

```
<?php
    //连接数据库
   $connection=mysql_connect("localhost","root","000");
  if(!connection)
  {
    die('连接数据库失败的原因是：'.mysql_error());
  }
  echo "当前使用的字符编码集为：".mysql_client_encoding($connection)."<br/>";
  printf("MySQL 服务器版本为：".mysql_get_server_info($connection)."<br/>");
  printf("连接方式：".mysql_get_host_info()."<br/>");
  printf("客户端版本：".mysql_get_client_info()."<br/>");
  printf("服务器状态为：".mysql_stat());
?>
```

(3) 新建 HTML 文件，命名为 9-9-index.html。

(4) 编辑 9-9-index.html，将前面案例中使用的 left.html、top.html 和 9-9.php 页面使用 Frameset 整合到一起。

9.9.4　运行结果

在浏览器中运行 9-9-index.html 页面，在页面中显示 MySQL 服务器信息和客户端信息，如图 9-45 所示。



图 9-45　获取 MySQL 服务器和客户端信息页面效果图

9.9.5　实例分析



源码解析：

在上面实例的 9-9.php 页面中，依次使用 mysql_client_encoding()函数获取了所使用的字符编码集；使用 mysql_get_server_info()函数获取 MySQL 服务器版本；使用 mysql_get_host_info()获取连接方式；使用 mysql_get_client_info()函数获取客户端版本号；使用 mysql_stat()函数获取服务器状态，并在页面中显示。

9.10　使用 PHP 准备语句完成学生信息管理系统

在项目开发时，通常会遇到一条 SQL 语句需要执行多遍，只是每次需要添加或者只是每次添加或修改的数据不同的情况，有时也会遇到查询的结果集中需要顺序颠倒一下再输出的情况，而这两种情况，使用 PHP 的准备语句都可以完成。本节将介绍准备语句。



视频教学：光盘\videos\09\准备语句.avi　　长度：6 分钟

9.10.1　基础知识——准备语句

在 MySQL 引入了两种准备语句来解决重复执行查询带来的开销大、编写代码不方便的问题，分别是绑定参数准备语句和绑定结果准备语句。

1. 绑定参数准备语句

绑定参数准备语句允许将查询存储在 MySQL 服务器上，只将迭代数据重复地发送给服务器，再将这些迭代数据集成到查询中执行。例如，有一个需要录入学生档案的 Web 应用程序，可以创建一个 Web 表单，接受最多 10 个学生的编号、姓名和性别，由于此信息使用相同的 SQL 语句插入数据库中，所以可以使用绑定参数准备语句。使用绑定参数准备语句执行 SQL 语句时，用到的方法有 mysqli_stmt_prepare()方法、mysqli_stmt_execute()方法、mysqli_stmt_close()方法和 mysqli_stmt_bind_param()方法。这些方法的介绍如表 9-1 所示。

表 9-1　准备语句方法

方法名称	过程化语法格式	参数描述	功能说明
mysqli_stmt_prepare()	mysqli_stmt_prepare(mysqli_stmt stmt , string query)	stmt 表示一个准备语句对象；query 表示一条 SQL 语句	准备要执行的语句
mysqli_stmt_execute()	mysqli_stmt_execute(mysqli_stmt stmt)	stmt 表示一个准备语句对象	执行准备语句，该语句何时执行，取决于语句类型
mysqli_stmt_close()	mysqli_stmt_close(mysqli_stmt stmt)	同上	关闭准备语句占用的资源
mysqli_stmt_bind_param()	mysqli_stmt_bind_param(mysqli_stmt stmt, string types , mixed &var1[, mixed &...])	stmt 表示一个准备语句对象；types 参数表示其后各个变量($var1...$varn)的数据类型，types 绑定的变量的数据类型接受 4 个参数：I 表示 integer 类型；D 表示 double 类型；S 表示 string 类型；B 表示 blob 类型	将变量名绑定到相应的字段(绑定参数)

由表 9-1 可知，mysqli_stmt_prepare()方法表示准备要执行的语句，有一点要说明，无论使用绑定参数还是使用绑定结果的准备语句，都需要使用该方法准备要执行的语句。下面创建一个实例，具体说明绑定参数如何执行 SQL 语句，代码如下：

```
<?php
    $link = mysqli_connect('localhost', 'root', '000', 'db_student');//连接数
据库
    //判断连接是否成功
    if (!$link) {
        printf("Connect failed: %s\n", mysqli_connect_error());
        exit();
    }
    $stmt = mysqli_prepare($link, "INSERT INTO student VALUES (?, ?, ?, ?,?)");
```

```
    mysqli_stmt_bind_param($stmt, 'isiss', $id, $name, $age, $sex,$address);
    //绑定参数
    $id = 1010;
    $name = '杨芳';
    $age = 22;
    $sex = "female";
    $address="北京市";
    $stmt->execute();   //执行预准备语句
    $stmt->close();
    mysqli_close($link);
?>
```

在上面的代码中“?”代表一个占位符，即后面插入学生信息的学号、姓名、年龄、性别和家庭住址。在下面调用$stmt->bind_param()方法，以方法中出现的相同顺序将变量$id、$name、$age、$sex 和$address 绑定到问号表示的字段占位符，并指定$stmt->bind_param()方法的第一个参数为“isiss”，表示要插入的数据类型依次为 integer、string、integer、string、string。此时每个数据行都已准备好，使用$stmt->execute()方法发送给服务器处理，最后，关闭并回收占用的资源。

2. 绑定结果准备语句

绑定结果准备语句允许将 PHP 变量绑定到所获取的相应字段，从而使有时难以处理的索引数组或关联从结果中提取值，然后在必要时使用这些变量。使用绑定结果准备语句执行 SQL 语句时，用到的方法有 mysqli_stmt_prepare()、mysqli_stmt_bind_result 和 mysqli_stmt_fetch()方法，由于第一个方法在表 9-1 中已经介绍，在此不再赘述，下面介绍 mysqli_stmt_bind_result()和 mysqli_stmt_fetch()方法。

mysqli_stmt_bind_result()方法将变量绑定到所获取的字段上，该函数的语法格式如下：

```
bool mysqli_stmt_bind_result ( mysqli_stmt stmt, mixed &var1 [, mixed &...] )
```

mysqli_stmt_fetch()方法将所获取的字段绑定到一个变量上，该函数的语法格式如下：

```
bool mysqli_stmt_fetch ( mysqli_stmt stmt )
```

下面创建一个实例，使用绑定结果准备语句查询数据，代码如下：

```
<?php
    $mysqli=new mysqli("localhost","root","000","db_student");  //实例化 mysqli
    $query="select * from student";
    $result=$mysqli->prepare($query);                         //进行预准备语句查询
    $result->execute();                                       //执行预准备语句
    $result->bind_result($id,$name,$age,$sex,$address);       //绑定结果
    while ($result->fetch()) {
            echo $id." ";
            echo $name." ";
            echo $age." ";
            echo $sex." ";
            echo $address."<br/>";
    }
```

```
    $result->close();                                          //关闭预准备语句
    $mysqli->close();                                          //关闭连接
?>
```

在该段代码中，查询语句中字段要和$result->bind_result()方法绑定参数的个数和顺序相同。$result->fetch()方法主要获取准备语句结果的每条记录，并将相应字段赋给绑定结果。在浏览器中查看运行结果，在页面中输出：

```
1002 马向林 23 male 河南省郑州市
1003 殷国鹏 22 male 河南省安阳市
```

9.10.2 实例描述

在前面案例中使用了 PHP 的一些内置函数完成了学生信息管理系统，那么，通过这一节的学习，可以用 PHP 的一些准备语句使学生信息管理系统的实现变得更加简单，下面就来修改学生信息管理系统，使之用 PHP 的一些准备语句来实现。

9.10.3 实例应用

【例 9-8】 使用 PHP 准备语句完成学生信息管理系统

(1) 新建 PHP 文件，命名为 9-10.php。

(2) 编辑 9-10.php，在页面中编辑添加学生信息的表单域，内容如下：

```
<form name="myForm" method="post" action="doAdd.php">
             <table cellpadding="0" cellspacing="0" border="1" width="95%"
style="margin-top:-50px;">
                 <tr>
                     <td>学号</td>
                     <td><input type="text" name="txtId" /></td>
                 </tr>
                 <tr>
                     <td>姓名</td>
                     <td><input type="text" name="txtName" /></td>
                 </tr>
                 <tr>
                     <td>年龄</td>
                     <td><input type="text" name="txtAge"></td>
                 </tr>
                 <tr>
                     <td>性别</td>
                     <td>
                         <select name="sSex">
                             <option value="male">male</option>
                             <option value="female">female</option>
                         </select>
                     </td>
```

```
                <tr>
                    <td>地址</td>
                    <td><input type="text" name="txtAddress"/></td>
                </tr>
                <tr>
                    <td colspan="2" align="center">
                        <input type="submit" name="submit" value="添加"/>
                        <input type="reset" name="reset" value="重置">
                    </td>
                </tr>
            </table>
</form>
```

(3) 当用户输入学生信息记录后，单击页面 9-10.php 页面中的【添加】按钮后，表单提交至 doAdd.php 文件，在该文件中处理表单数据，并执行添加操作，将用户输入的学生信息添加到数据表中。代码如下：

```
<?php
    $mysqli=new mysqli("localhost","root","000","db_student");//获取连接数据库
    $id=$_POST['txtId']|null;
    $name=$_POST['txtName'];
    $age=$_POST['txtAge'];
    $sex=$_POST['sSex'];
    $address=$_POST['txtAddress'];
    $query="insert into student values(id=?,name=?,age=?,sex=?,address=?)";
        //插入 SQL 语句
    $stmt=$mysqli->stmt_init();
    $stmt->prepare($query);//准备要执行的语句
    $stmt->bind_param('isiss',$id,$name,$age,$sex,$address);
        //向 SQL 语句添加参数，绑定参数
    $stmt->execute();   //执行预准备语句
    $stmt->close();
    $mysqli->close();
    header("location:stuList.php");
?>
```

(4) 当执行完添加数据操作后，页面自动跳转至 stuList.php 页面，该页面会将数据表中的所有记录查询并显示在页面中。编辑 stuList.php 页面，在页面中使用 PHP 的准备语句查询数据表中的所有学生信息，代码如下：

```
<?php
    $mysqli=new mysqli("localhost","root","000","db_student");//实例化 mysqli
    $query="select * from student";
    $result=$mysqli->prepare($query);                        //进行预准备语句查询
    $result->execute();                                      //执行预准备语句
    $result->bind_result($id,$name,$age,$sex,$address);      //绑定结果
    echo "<table width='95%'><tr>";
    echo "<td>学号</td><td>姓名</td><td>年龄</td><td>性别</td><td>住址</td>";
```

```
    echo "</tr>";
    while ($result->fetch()) {
            echo "<tr>";
            echo "<td>".$id."</td>";
            echo "<td>".$name."</td>";
            echo "<td>".$age."</td>";
            echo "<td>".$sex."</td>";
            echo "<td>".$address."</td>";
            echo "</tr>";
    }
    echo "</table>";
    $result->close();          //关闭预准备语句
    $mysqli->close()           //关闭连接
?>
```

(5) 到此学生信息管理系统的添加、查询功能就完成了，保存 stuList.php 页面。

(6) 新建 HTML 文件，命名为 9-10-index.htm。

(7) 在 9-10-index.htm 页面中使用 Frameset 将前面案例中的 header.htm、menu.htm 和本节案例中的 9-10.php 页面整合。

9.10.4 运行结果

在浏览器中运行 9-10-index.htm，在页面中显示添加学生信息的表单域，如图 9-46 所示。



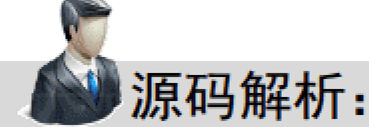
图 9-46 添加学生信息页面效果图

当用户输入完整学生信息后，单击页面中的【添加】按钮，执行添加操作，并查询数据表中的所有学生信息记录，以表格的形式在页面中显示，如图 9-47 所示。



图 9-47　学生信息列表页面

9.10.5　实例分析



源码解析：

在上面实例中，当用户输入完整学生信息后提交表单至 doAdd.php 页面中，在该页面中首先使用了 mysqli_prepare()函数准备要执行的 SQL 语句，然后使用 mysqli_stmt_bind_param()函数绑定参数，最后使用$stmt->execute()语句执行插入的 SQL 语句，关闭连接。页面自动跳转至 stuList.php 页面中，在该页面中，当执行完查询操作后，使用 mysqli_stmt_bind_result()函数绑定结果字段，并以表格的形式输出查询结果。

9.11　使用事务处理完成学生记录的删除操作功能

事务是作为一个单元的、一组有序的数据库操作。如果组中的所有操作都成功，则认为事务成功，即使只有一个操作失败，事务也会不成功。如果所有的操作都成功则事务提交(commit)，将作用于其他数据库进程，如果一个操作失败，则事务回滚(rollback)，该事务所有影响将被取消。本节将详细介绍 PHP 中的事务处理函数。

9.11.1　网络教学

视频教学：光盘\videos\09\事务处理.avi　　　　长度：9 分钟

9.11.2　基础知识——事务处理

事务是数据库的实现，而非语言的实现，语言的实现只不过是包装了数据库提供的 AIP 或者直接使用数据库所提供的事务相关的 SQL Statement，在 PHP 5.0 中可以调用 MySQL 的 API 来实现事务的处理。PHP 主要通过下列函数实现事务处理，分别为 autocommit()、commit()和 rollback()。下面详细介绍这些函数。

1．控制 MySQL 自动提交模式的行为

autocommit()函数控制 MySQL 自动提交模式的行为，该函数的过程化语法格式如下所示：

```
bool autocommit(boolean mode)
```

autocommit()函数面向对象语法格式如下：

```
class mysqli {
    bool autocommit ( bool mode )
}
```

2．将当前事务提交给数据库

commit()函数将当前事务提交给数据库，成功时返回 true，否则返回 false。该函数的语法格式如下：

```
bool commit()
```

该函数面向对象语法格式如下：

```
class mysqli {
    bool commit ( void )
}
```

3．回滚当前事务

rollback()函数表示回滚当前事务，成功时返回 true，否则返回 false。该函数的过程化语法格式如下：

```
rollback()
```

该函数面向对象的语法格式如下所示：

```
class mysqli {
    bool rollback ( void )
}
```

9.11.3　实例描述

在上一节的实例中已经使用准备语句完成了学生信息管理系统的添加和查询操作，下面来进一步完善该系统，实现学生记录的删除操作。

9.11.4　实例应用

【例 9-9】 使用事务处理完成学生记录的删除操作功能

(1) 在上面案例的 stuList.php 页面表格中，添加一行名称为“操作”的列，并在表格的每行添加【删除信息】超链接，代码如下：

```
<td><a href='9-11.php?id=".$id."'>【删除信息】</a></td>
```

当单击【删除信息】超链接时，页面跳转至 9-11.php 页面，并将该条记录的 id 传递到 9-11.php 页面中。

(2) 新建 PHP 文件，命名为 9-11.php。

(3) 编辑 9-11.php，获取连接地址中的参数 id 值，使用事务处理完成删除操作，代码如下：

```
<?php
    $con=new mysqli('localhost','root','000','db_student');    //初始化 mysqli
    if(!$con)
    {
        echo mysqli_error();
    }
    $success=true;
    $con->autocommit(false);
    $query="delete from student where id=".$_GET['id'];    //删除的 SQL 语句
    $result=$con->query($query);                           //执行删除
    if(!$result)
    {
        $success=false;
    }
    if($success)
    {
        $con->commit();
        echo "事务提交成功，已成功删除一条信息！";
    }
    else
    {
        $con->rollback();
        echo "事务提交失败！";
    }
    $con->autocommit(true);
?>
```

9.11.5　运行结果

运行 9-10-index.htm 页面，添加一条信息到数据表中，然后显示添加后的数据表中所有的

记录信息。当用户选中一条记录，并单击该条记录后面的【删除信息】超链接时，页面跳转至 9-11.php 页面中，提示用户删除成功的消息，如图 9-48 所示。



图 9-48　事务处理完成删除操作

9.11.6　实例分析



源码解析：

在上面实例中，首先设置 MySQL 的自动提交模式为 false，然后执行删除操作，当用户执行删除成功后，使用 commit()函数提交事务，并提示用户“事务提交成功，已成功删除一条信息！”；当用户执行删除失败后，使用 rollback()函数将事务回滚，最后将事务自动提交，设置 autocommit 为 true。

9.12　常见问题解答

9.12.1　PHP 连接国外 MySQL 数据库问题



PHP 连接国外的 MySQL 数据库问题。
网络课堂：http://bbs.itzcn.com/thread-12469-1-1.html。

我用 PHP 连接国外的 MySQL 数据库，总是出现两个错误，不停地刷新，这两个错误就会交替出现。

错误一：

```
Lost connection to MySQL server during query
```

错误二：

```
Access denied for user 'u1047157_emamm'@'%' to database 'db1047157_emamm'
```

请问什么原因啊？下面是我连接 MySQL 的代码：

```
<?php
   $conn = mysql_connect("db720.mydomainwebhost.com:3306", "u1047157_pony-e",
"151012") or die(mysql_error());
    mysql_query("SET NAMES 'utf8'") or die(mysql_error());//编码为 UTF-8
    mysql_select_db("db1047157_emamm") or die(mysql_error());//打开指定的数据库
    $exec="select * from users_info" or die(mysql_error());
    $return=mysql_query($exec) or die(mysql_error());
    while($rs=mysql_fetch_object($return)){
        echo $rs->users_id;
    }
?>
```

【解决办法】 这个错误是数据库设置的问题，允许外部主机权限没开放。虽然用的不是自己的主机上的数据库，但是还是需要开放外部主机才行，还需要输入外部主机的 IP。

9.12.2　用 PHPmyAdmin 操作 MySQL 数据库出现乱码



用 PHPmyAdmin 操作 MySQL 数据库出现乱码。
网络课堂：http://bbs.itzcn.com/thread-12470-1-1.html。

用 PHPmyAdmin 操作 MySQL 数据库时，汉字显示正常，但是 PHP 网页显示 MySQL 数据库时所有汉字都变成了“?”号，这是怎么回事？怎么解决呢？

【解决办法】 在 PHP 连接 MySQL 完成后，在从数据表中取值之前加上如下代码。

```
mysql_query("SET NAMES xxx");
```

“xxx”要和页面上的编码一样就可以解决这个乱码问题了。

9.13　习　　题

一、填空题

(1) mysql_connect()函数的主要作用是打开或重复使用一个到 MySQL 服务器的非持久性连接，如果要创建持久性连接需要使用__________函数。

(2) 假设数据库已经连接成功，阅读下面的代码，如果需要获取数据表中“username”字段的值，划线处应当填写__________。

```
$sql = "SELECT * from tab_user";
//使用 mysql_query()函数执行 sql 语句
$result = mysql_query($sql,$con);
//输出查询结果中指定字段的值
echo "用户名："._________."<br/>";
```

(3) mysql_list_dbs()函数列出 MySQL 服务器中所有的数据库，__________函数取得 mysql_list_dbs()调用所返回的数据库名。

二、选择题

(1) 当 PHP 连接或操作 MySQL 出错时，获取错误信息的函数有两个，分别是__________。

A. mysql_error()和 mysql_errno()

B. mysql_errorMsg()和 mysql_error()

C. mysql_query()和 mysql_error()

D. mysql_error()和 mysql_err()

(2) 假设数据库已连接成功，数据表 tab_user 中有 5 个字段，分别是 id、username、password、sex、age，那么执行下面的代码，输出的结果是__________。

```
$sql = "SELECT username , password , sex , age FROM tab_user";
$result = mysql_query($sql,$con);//执行查询操作
echo mysql_num_fields($result);//输出表中字段数量
```

A. 5　　B. 4　　C. 3　　D. 2

(3) 假设数据库已连接成功，使用绑定结果获取数据表中的信息代码如下：

```
$query="select * from student";
________
$result->execute();                          //执行预准备语句
```

这段代码中，划线处应当填写以下哪项才得以执行成功。

A. $result=prepare($query);

B. $result->prepare($query);

C. $result=$mysqli->prepare($query);

D. $result=$mysqli=prepare($query);

三、上机练习

上机练习： 检测数据表中重复数据并删除。

新建企业后台管理首页页面，在页面中显示学生信息数据表中所有数据，查看是否有重复的数据，如图 9-49 所示。



图 9-49　学生信息管理系统首页

如果有重复的数据，单击数据列表右上角的【删除重复信息】超链接，页面跳转至

exercise.php 页面中，在该页面中查询重复的信息并删除，如图 9-50 所示。



图 9-50　检测出重复数据并删除



第 10 章　晚上关灯闭户防贼——PHP 安全技术

内容摘要：

计算机网络技术的飞速发展，大大改变了人们的生活面貌，促进了社会的发展。互联网是一个面向大众的开放系统，对于信息的保密和系统的安全性考虑得并不完备，由此引起的网络安全问题日益严重。在 PHP 的程序开发中，PHP 自带安全配置和数据加密，但为了适应更为复杂的数据破坏，我们还需要了解 PHP 强大的加密扩展库，在这一章里将进行详细地讲解。

学习目标：

- 掌握 PHP 中与安全有关的配置。
- 掌握 PHP 的信息隐藏策略。
- 掌握 PHP 中常见的加密技术。
- 了解和使用 PHP 的加密扩展库。

10.1　PHP 安全配置

程序员在实际应用开发中，无论从自身系统考虑还是从用户的实用性考虑，都需要注重程序的安全性。在这一节里我们将了解怎样安全地配置 PHP。

视频教学：光盘\videos\10\PHP 的安全配置.avi　　长度：12 分钟

10.1.1　基础知识——安全配置

PHP 是 Web 服务器的一个模块功能，所以首先要保证 Web 服务器的安全。这样我们就需要先了解 PHP 的安全模式和安全配置指令。

1. 安全模式

对于在共享服务器中运行的用户而言，启用 PHP 的安全模式是非常有用的。启用安全模式后，PHP 总是会验证用户的权限，因此只要为文件配置了正确的权限，就能防止用户无意修改而导致的系统错误。但是启用安全模式会对 PHP 的行为产生一些影响，甚至会禁锢一些标准 PHP 函数的功能。下面让我们来讲一下这些影响和安全模式有关的参数。

safe_mode(boolean)指令将限制在共享环境中使用 PHP 时可能有危险的语言特性。作用域是 PHP_INI_SYSTEM，默认为 0。在执行 PHP 时，它会执行脚本的 UID 和脚本尝试访问的文件的 UID。以此来作为限制的基础。实际应用中，当启用该安全模式时，以下 5 项限制将生效。

- 如果试图创建新文件，将限制为只能在该用户拥有的目录中创建。
- 所以输出和输入函数将受到限制，只能拥有与调用这些函数的脚本用相同拥有者的文件。
- 如果试图通过 popen()、system()或者 exec()等函数执行脚本，只能在安全模式指令 safe_mode_exec_dir 指定的目录下才能执行。
- 如果使用 MySQL 数据库服务器，连接服务器的用户名必须和调用 mysql_connect()函数的文件的拥有者的用户名相同。
- 将认证脚本拥有者的 UID 划入认证领域范围内，使 HTTP 认证得以加强。当启用安全模式，将不会启用 PHP_AUTH。

启用 safe_mode(boolean)函数时，受到影响的所有变量、函数和配置指令如表 10-1 所示。

表 10-1　启用 safe- mode 指令时受影响的函数变量及配置指令

apache_request_headers()	Dbmopen()	Highlight_file()	Mysql_*
Backticks()	Dl()	Ifx_*	Symlink()
Chdir()	Exec()	Ingres_*	Unlink()

续表

Chgrp()	Filepro()	Link()	Shell_exec()
Chmod()	Filepro_retrieve()	Mail()	System()
Chown()	Filepro_rowcount()	Max_execution_time()	Rmdir()
Copy()	Fopen()	Mkdir()	Putenv()
Dbase_open()	Header()	Move_uploaded_file()	Touch()

safe_mode 指令是在 PHP 中启用安全模式的指令，当启用安全模式后，我们还需要安全模式下的其他指令来进行更安全的配置。这些指令如表 10-2 所示。

表 10-2　安全模式下的指令

指　令	作 用 域	默 认 值	详细说明
safe_mode_gid(boolean)	PHP_INI_SYSTEM	0	该指令会修改安全模式的行为，即从执行前验证 UID 改为验证组 UID
safe_mode_include_dir (string)	PHP_INI_SYSTEM	Null	该指令可以指定多个路径，当启用安全模式时，在这些路径中将忽略安全模式
safe_mode_allowed_env_ vars(string)	PHP_INI_SYSTEM	"PHP_"	当启用安全模式时，可以使用此指令允许用户的脚本修改某些环境变量。可以修改多个变量，变量之间用分号分隔
safe_mode_exec_dir(string)	PHP_INI_SYSTEM	Null	该指令指定一些目录，其中的程序可以使用 system()、exec() 或 passthru() 函数执行，在系统中必须用(/)作为分隔符
safe_mode_protected_env_ vars(string)	PHP_INI_SYSTEM	LD_LIBRARY_PATH	该指令用于保护某些环境变量不被 putenv() 函数修改，默认情况下，变量 LD_LIBRARY_PATH 是受保护的，因为如果在运行时修改这个变量，可能导致不可预知的结果

当使用 safe_mode_include_dir(string)函数指定某个路径但未包含最后的斜线，则该路径下的所用目录都会忽略安全模式设置。例如，如果设置此指令为/localhost/configuration，则表示/localhost/configuration/templates 和/localhost/configuration/passwords/都将排除在安全模式之外。所以，如果只要排除一个或者多个目录不受安全模式的限制，要确保每个目录都要包括斜线。

2. 配置参数

除与 PHP 安全有关的处理 PHP 的安全模式外，还用到其他的一些用于的配置参数，这些配置参数，能够在你的 PHP 安全中起到重要的作用。这些配置参数的详细信息如表 10-3 所示。

表 10-3　安全配置参数

指　令	作 用 域	默 认 值	详细说明
disable_functions(string)	PHP_INI_SYSTEM	NULL	该指令用于禁用你希望禁用的函数名列表，不依赖于是否启用安全模式
disable_classes(string)	PHP_INI_SYSTEM	NULL	该指令用于禁用你希望禁用的类名列表，不依赖于是否启用 safe_mode() 命令
doc_root(string)	PHP_INI_SYSTEM	NULL	该指令用于设置一个路径，指定提供 PHP 文件的根目录，如果默认为空，则忽略，并将按照 URL 所指定的执行脚本
max_execution_time(integer)	PHP_INI_ALL	30	该指令用于指定脚本在中止前的执行秒数，对于防止用户脚本占用过多 CPU 时间非常的有用，默认为 0 时，则没有时间限制
memory_limit(integer)	PHP_INI_ALL	8M	该指令用于指定脚本使用的内存大小，必须以 MB 为单位，不能指定其他的值，并且必须在数字后面加 M，此指令在启用—enable-memory-limit 后才能使用
open_basedir(string)	PHP_INI_SYSTEM	NULL	该指令用于创建一个基目录，将限制所用的文件只能在这个文件中进行，因此可以防止用户进入服务器的受限区域，不依赖于是否启用 safe_mode()命令

续表

指　令	作 用 域	默 认 值	详细说明
sql.safe_mode(integer)	PHP_INI_SYSTEM	0	启用该指令，会忽略传给 mysql_connect()和 mysql_pconnect()的所有信息，而使用 localhost 作为目标主机，运行 PHP 的用户作为用户名而不适用密码
user_dir(string)	PHP_INI_SYSTEM	NULL	该指令指定用户主目录的一个用户名，规定 PHP 脚本必须在这里才能执行

当使用 open_basedir(string)函数时，假如我们编写的所用的程序和素材都在目录/localhost/www 下，在此我们可使用如下设置来防止用户通过 PHP 指令浏览和操作文件，代码如下：

```
open_basedir = "/localhost/www";
```

10.1.2　实例描述

我们在开发时，经常会处理一些有关密码或者重要信息的文件，这些信息都是我们在编写程序时要考虑保护的，下面我们来制作一个实例，来防止攻击者攻击和操作文件。

10.1.3　实例应用

【例 10-1】　文件的安全上传

(1)　启用 PHP 安全模式命令。

(2)　新建一个文件，命名为 11.1.php，在页面中我们实现一个能够上传文件的 form 表单，代码如下：

```
<title>文件上传</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</head>
<body bgcolor="#FFFFFF">
<form enctype="multipart/form-data" method="post">
上传文件：
<input type="file"  name="file" size="30">
<input type="submit" name="upload" value="上传">
</form>
```

(3)　我们为验证上传添加验证代码，其代码如下：

```
<?php
if(isset($upload) && $file != "none") {
```

```
    copy($file, "d:/www/upload/".$file_name);
    echo "文件".$file_name."上传成功! 单击<a href=\"$PHP_SELF\">继续上传</a>";
    exit;
}
?>
```

(4) 保存文件。

10.1.4 运行结果

将文件在浏览器中运行，选一种文件进行上传，我们可以看如此编程所存在的一个重大的错误，如下所示：

```
Warning: copy(d:/www/upload/index.html) [function.copy]: failed to open stream:
No such file or directory in D:\www\test\11.1.php on line 5
```

根据这个错误，攻击者就可以通过访问以下路径。

```
http://localhost/www/ upload/index.html
```

得到这个文件，泄露脚本源码，攻击者可以自定义 form 里 file_name 变量的值，上传覆盖任意有写权限的文件，还可以上传 PHP 脚本执行主机的命令。

以上的错误处理是我们不希望看到的，这样容易使我们的文件泄露，很不安全。PHP 提供了 is_uploaded_file 和 move_uploaded_file 函数，可以检查操作的文件是否是用户上传的文件，从而避免把系统文件复制到 web 目录。使用$HTTP_POST_FILES 数组来读取用户上传的文件变量。严格检查上传变量，比如不允许是 PHP 脚本文件。

把 PHP 脚本操作限制在 web 目录可以避免程序员使用 copy 函数把系统文件复制到 web 目录。move_uploaded_file 不受 open_basedir 的限制，所以不必修改 php.ini 里 upload_tmp_dir 的值。

10.1.5 实例分析



源码解析：

上述实例中我们首先做了一个不安全的文件上传的例子，在例子中发现很容易泄露我们的服务器文件路径，这对一个 Web 项目来说是致命的，我们可以使用 move_uploaded_file 指令使我们要上传的文件不用复制到我们的服务器 Web 目录。

10.2 采用隐藏策略

很多 PHP 程序员都倾向于开源软件，但是对于用开源软件来编程，我们在项目的发布时，项目的每一项信息都有可能为攻击者提供一些重要的线索，最终使我们的服务器被入侵。因此，程序员要在编程时学会隐藏策略，隐藏一些技术和配置细节或者敏感的数据等。

视频教学：光盘\videos\10\采用隐藏策略.avi　　　　长度：8 分钟

10.2.1　基础知识——隐藏策略

PHP 的隐藏策略主要包括 PHP 配置信息细节的隐藏和敏感数据的隐藏。下面我们来看看这两种隐藏策略。

1. Apache 和 PHP 配置信息细节的隐藏

在 Apache 服务器中，所用文档的请求和服务器生成的文档中都会输出一个服务器签名，控制这项签名的指令有两个，分别是 ServerSignature 和 ServerTokens 指令。

- ServerSignature 指令主要负责 Apache 服务器的版本、服务器名、端口和编译模块有关的一行输出。启用这个指令时，我们可以看到以下的输出：

```
Apache/2.2.14(Win32) DAV/2 PHP/5.2.8 Server at www.example.com port 80
```

- ServerTokens 指令主要是确定在启用 ServerSignature 指令时，以何种程度提供服务器的细节。一共有 6 个选项，如表 10-4 所示。

表 10-4　ServerTokens 指令选项

指令选项	例　子
Full	Apache/2.2.14(Win32) DAV/2 PHP/5.2.8
Minimal	Apache/2.2.14
Minor	Apache/2.0
OS	Apache/2.2.14(Win32)
Major	Apache/2
Prod	Apache

除此之外用其他的指令和方法也能够隐藏细节。

- Expose_php 指令，启用时将细节追加到服务器签名的后面。例如如果启用了 ServerSignature 命令，并将 ServerTokens 的值设为 Full，然后启用 Expose_php 命令，则服务器的签名就如下所示：

```
Apache/2.2.14(Win32) DAV/2 PHP/5.2.8 Server at www.example.com port 80
```

  如果禁用了 Expose_php 命令，则服务器的签名就为：

```
Apache/2.2.14(Win32) DAV/2 Server at www.example.com port 80
```

- 删除 phpinfo()调用的所用实例

  phpinfo()函数主要用于在指定的服务器上查看 PHP 配置信息的总结，包括操作系统的信息、PHP 和 Web 服务器版本的信息和 PHP 配置标志的有关信息，另外还有所有 PHP 扩展的版本的详细报告。如果这些信息让攻击者知道的话，那将给我们的系统带来不小的麻烦，他们可以查看服务器的许多技术特性和技术缺陷。

- 修改文档扩展名

  跟 PHP 接触时间长的人都知道，PHP 可以有很多的扩展名，如.php、.php3 和.phtmle 等。但是，可能还有人不知道 PHP 中可以很容易地修改你自己希望的扩展名，甚至是向.jsp 这样的也可以。我们只需要在 httpd.conf 文件中配置一下就行了。在 httpd.conf 文件中找到如下的一行代码：

```
AddType application/x-httpd-php .php
```

  我们可以修改成想要的扩展名，如下代码所示：

```
AddType application/x-httpd-php .hphp
```



设置的扩展名一定要确保与其他安装的服务器技术不相冲突。

2. 敏感数据的隐藏

事实上，对于任何 Web 服务器文档树中的文档，只要有足够的权限，就可以通过任何能执行 GET 命令的机制来获取文档中的信息。即使你已经修改了一个很奇怪的扩展名，那对于攻击者来说只是费一些时间而已，一旦知道了你的文件名，他们就可以大肆访问你的文件内容了。考虑到上述问题，我们有两种方法解决。

- 注意文档和目录：要把重要文件放在公共 HTML 文档目录路径之外的路径之中。用到时，只需使用 include()函数，将文件包含在 PHP 文件中即可。但是，如果禁用了安全模式，能够执行 PHP 脚本的其他用户仍能将此文件包含到自己的文件中，因此在注意文档和目录的同时，在共享主机中，还要启用安全模式形成双重保护。
- 设置拒绝访问某些文件的扩展名，在 httpd.conf 文件中修改如下代码。

```
<FilesMatch  *.ini>
   Order allow,deny
   Deny from all
</FilesMatch>
```

这样再重新启动服务器，然后任何的用户在浏览器中访问带.ini 扩展名的文件，都将被拒绝。

10.2.2 实例描述

一般而言，通过隐藏的手段提高安全性被认为是作用不大的。但某些情况下，尽可能多增加一份安全性都是值得的。一些简单的方法可以帮助隐藏 PHP，这样做可以提高攻击者发现系统弱点的难度。这个实例中我们将讲解如何通过隐藏 PHP 的扩展名来提高 PHP 的安全性。

10.2.3 实例应用

【例 10-2】 隐藏 PHP 的扩展名

(1) 新建一个名为 11.2.py 的文件，使用.py 作为 php 文件的隐藏扩展名，在文件中写入如下 PHP 代码，我们使用 phpinfo()来输出 PHP 信息。

```
<?php
 echo "我是隐藏的 php";
 phpinfo();
?>
```

(2)　在 Apache 服务器的安装目录 D:\Program Files\Apache Software Foundation\Apache2.2\conf 中找到 httpd.conf 文件，打开并找到 AddType application 字段，然后添加以下代码：

```
AddType application/x-httpd-php .asp .py .pl
```

(3)　保存文件，重启 Apache 服务器。

10.2.4　运行结果

打开浏览器，在地址栏中输入 http://localhost/test/11.2.py，单击【转到】按钮，我们可以看到，以.py 为扩展名的文件也可以访问，效果如图 10-1 所示。



图 10-1　访问.py 文件

10.2.5　实例分析



源码解析：

上述实例中我们在 httd.conf 文件中使用 AddType application/x-httpd-php 把 PHP 的扩展名修改成了隐藏的.py 文件。这里值得注意的是，我们修改的 httd.conf 文件必须在 Apache 服务器重启后，隐藏的扩展名才能够生效。

10.3 认识加密技术

随着网络时代的高速发展，我们越来越认识到信息对每一个人的重要性。怎样能使我们的重要信息安全地在网络间传送，已经成为程序员们重要的研究方向。而这一研究的主要技术就是对数据的加密技术。

视频教学：光盘\videos\10\加密技术.avi　　长度：8 分钟

10.3.1 基础知识——加密技术

加密技术是 PHP 中采取的主要安全保密措施，是最常用的安全保密手段，利用技术手段把重要的数据变为乱码(加密)传送，到达目的地后再用相同或不同的手段还原(解密)。

对数据的加密技术分为两个方面，对称加密和非对称加密，如表 10-5 所示。

表 10-5　数据加密技术

加密技术	典型算法代表	解密方法
对称加密(私人密钥加密)	数据加密标准(DES，Data Encryption Standard)算法	对称加密的加密密钥和解密密钥相同
非对称加密(公开密钥加密)	RSA(Rivest Shamir Adleman)算法	非对称加密的加密密钥和解密密钥不同，加密密钥可以公开而解密密钥需要保密

对称加密采用了对称密码编码技术，它的特点是文件加密和解密使用相同的密钥，即加密密钥也可以用作解密密钥，这种方法在密码学中叫做对称加密算法。对称加密算法使用起来简单快捷，密钥较短，且破译困难，除了数据加密标准(DES)。另一个对称密钥加密系统是国际数据加密算法(IDEA)，它比 DES 的加密性好，而且对计算机功能要求也没有那么高。

非对称加密算法需要两个密钥：公开密钥和私有密钥。公开密钥与私有密钥是一对，如果用公开密钥对数据进行加密，只有用对应的私有密钥才能解密；如果用私有密钥对数据进行加密，那么只有用对应的公开密钥才能解密。因为加密和解密使用的是两个不同的密钥，所以这种算法叫做非对称加密算法。

当然，加密技术的应用是多方面的，但最为广泛的还是在电子商务和 VPN 上的应用，下面就分别简单介绍。

1. 在电子商务方面的应用

电子商务要求顾客可以在网上进行各种商务活动，不必担心自己的信用卡会被人盗用。过去，用户为了防止信用卡的号码被窃取到，一般是通过电话订货，然后使用用户的信用卡进行付款。现在人们开始用 RSA(一种公开/私有密钥)加密技术，提高信用卡交易的安全性，从而使电子商务走向实用成为可能。

许多人都知道 Netscape 公司是 Internet 商业中领先技术的提供者，该公司提供了一种基于

RSA 和保密密钥的应用于因特网的技术，被称为安全插座层(Secure Sockets Layer，SSL)。也许大家对 Socket 不陌生，它是一个编程界面，并不提供任何安全措施，而 SSL 不但提供编程界面，而且向上提供一种安全的服务，SSL3.0 现在已经应用到了服务器和浏览器上，SSL2.0 则只能应用于服务器端。

SSL3.0 用一种电子证书(electric certificate)来实行身份进行验证后，双方就可以用保密密钥进行安全的会话了。它同时使用“对称”和“非对称”加密方法，在客户与电子商务的服务器进行沟通的过程中，客户会产生一个 Session Key，然后客户用服务器端的公钥将 Session Key 进行加密，再传给服务器端，在双方都知道 SessionKey 后，传输的数据都是以 Session Key 进行加密与解密的，但服务器端发给用户的公钥必需先向有关发证机关申请，以得到公证。

基于 SSL3.0 提供的安全保障，用户可以自由订购商品并且给出信用卡号，也可以在网上和合作伙伴交流商业信息并且让供应商把订单和收货单从网上发过来，这样可以节省大量的纸张以及电话和传真费用。过去，电子信息交换(Electric Data Interchange，EDI)、信息交易(information transaction)和金融交易(financial transaction)都是在专用网络上完成的，使用专用网的费用大大高于互联网。正是这样巨大的诱惑，才使人们开始发展因特网上的电子商务，且进行数据加密。

2．加密技术在 VPN 中的应用

现在，越来越多的公司走向国际化，一个公司可能在多个国家都有办事机构或销售中心，每一个机构都有自己的局域网 LAN(Local Area Network)，但在当今的网络社会用户的要求不仅如此，还希望将这些 LAN 联结在一起组成一个公司的广域网，这个现在已不是什么难事了。

事实上，很多公司都已经这样做了，但他们一般使用租用专用线路来联结这些局域网，他们考虑的就是网络的安全问题。现在具有加密/解密功能的路由器已到处都是，这就使人们通过互联网连接这些局域网成为可能，这就是我们通常所说的虚拟专用网(Virtual Private Network ，VPN)。当数据离开发送者所在的局域网时，该数据首先被用户端连接到互联网上的路由器进行硬件加密，数据在互联网上是以加密的形式传送的，当达到目的 LAN 的路由器时，该路由器就会对数据进行解密，这样目的 LAN 中的用户就可以安全看到真正的信息了。

10.3.2　实例描述

以下假设你是一个中级 PHP Web 程序开发人员，要创建一个登录页面，在页面中的表单中，要求输入用户名和密码，来看你该如何实现。

10.3.3　实例应用

【例 10-3】 初次使用加密密码

(1) 新建一个名为 11.3.htm 的文件，在文件中我们添加含有账号和密码的表单代码如下：

```
<FORM name=form1 action=index.php method=post>
              <TBODY>
              <TR>
                <TD class=td align=right width="33%">账 号:</TD>
```

```
            <TD class=td width="67%"><INPUT
              style="BORDER-RIGHT: #0d5c95 1px solid; BORDER-TOP: #0d5c95 1px
solid;  BACKGROUND:  #fff;  BORDER-LEFT:  #0d5c95  1px  solid;  WIDTH:  140px;
BORDER-BOTTOM: #0d5c95 1px solid; HEIGHT: 21px"
              name=loginname> </TD></TR>
          <TR>
            <TD class=td align=right width="33%">密 码:</TD>
            <TD class=td width="67%"><INPUT
              style="BORDER-RIGHT: #0d5c95 1px solid; BORDER-TOP: #0d5c95 1px
solid;  BACKGROUND:  #fff;  BORDER-LEFT:  #0d5c95  1px  solid;  WIDTH:  140px;
BORDER-BOTTOM: #0d5c95 1px solid; HEIGHT: 21px"
              type=password size=21 name=loginpwd> </TD></TR><!---->
          <TR>
            <TD class=td width="33%"> </TD>
            <TD class=td width="67%"><INPUT type=image height=33 width=83
              src="11.3_files/index_hz04.gif" name=Submit>
          </TD></TR></FORM>
```

(2) 新建一个 PHP 的表单验证页面，名为 index.php。在页面中添加一些代码：

```
<BODY bgColor=#006599 leftMargin=0 topMargin=0 marginwidth="0" marginheight=
"0">
<?php
session_start();
$conn= mysqli_connect("localhost","root","root","userman");
$conn->query('set names gb2312');

if(empty($conn))
    {
      die("数据库连接失败");
    }
$logname = $_POST['loginname'];
$logpass = $_POST['loginpwd'];
 $str="select * from admin where name='".$logname."'";
      $result1=$conn->query($str);
      $row=$result1->fetch_row();
if($row){
          $temp="已有人登录，请重新选择名字!";
          echo $temp;
          echo"<a href=11.3.htm>返回</a>";
    }else{
?>

<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD height=600><BR><BR><BR><BR>
      <TABLE class=tableborder cellSpacing=0 cellPadding=0 width=427
      align=center border=0>
        <TBODY>
```

```
            <TR>
              <TD background=11.3_files/index_hz01.gif colSpan=3
                height=64>  </TD></TR>
            <TR>
              <TD style="FONT-SIZE: 1px; LINE-HEIGHT: 1px" width=3
              background=11.3_files/index_hz02.gif></TD>
              <TD style="BORDER-TOP: #666666 1px double" vAlign=top
              background=11.3_files/index_hz03.gif height=162><BR><BR><!--
-->
                <TABLE id=adminlogin cellSpacing=0 cellPadding=0 width="100%"
                align=center border=0>
         <h2><?echo "登录成功";}?></h2>
                  </TABLE><!----></TD>
              <TD style="FONT-SIZE: 1px; LINE-HEIGHT: 1px" width=3
              background=11.3_files/index_hz02.gif></TD></TR>
            <TR>
              <TD background=11.3_files/index_hz05.gif colSpan=3
                height=127> 
</TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></BODY>
```

(3)　保存文件。

10.3.4　运行结果

将登录文件 11.3.htm 在服务器中运行，我们可以看到如图 10-2 所示的效果。

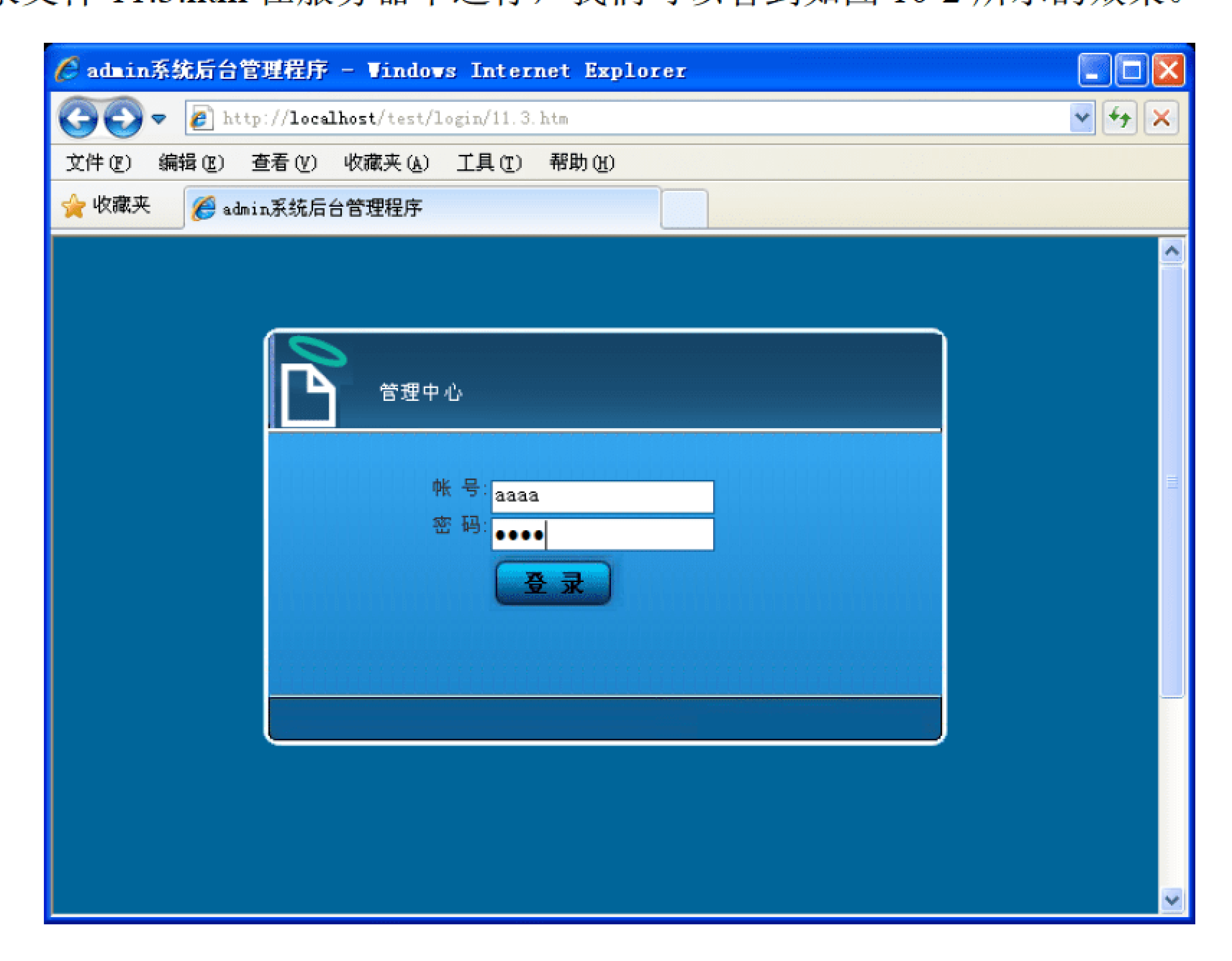


图 10-2　后台登录页面

在这个页面中我们可以看到，当输入密码时，密码的字符会以黑点代替，单击登录，我们会看到跳转处理页面如图 10-3 所示。

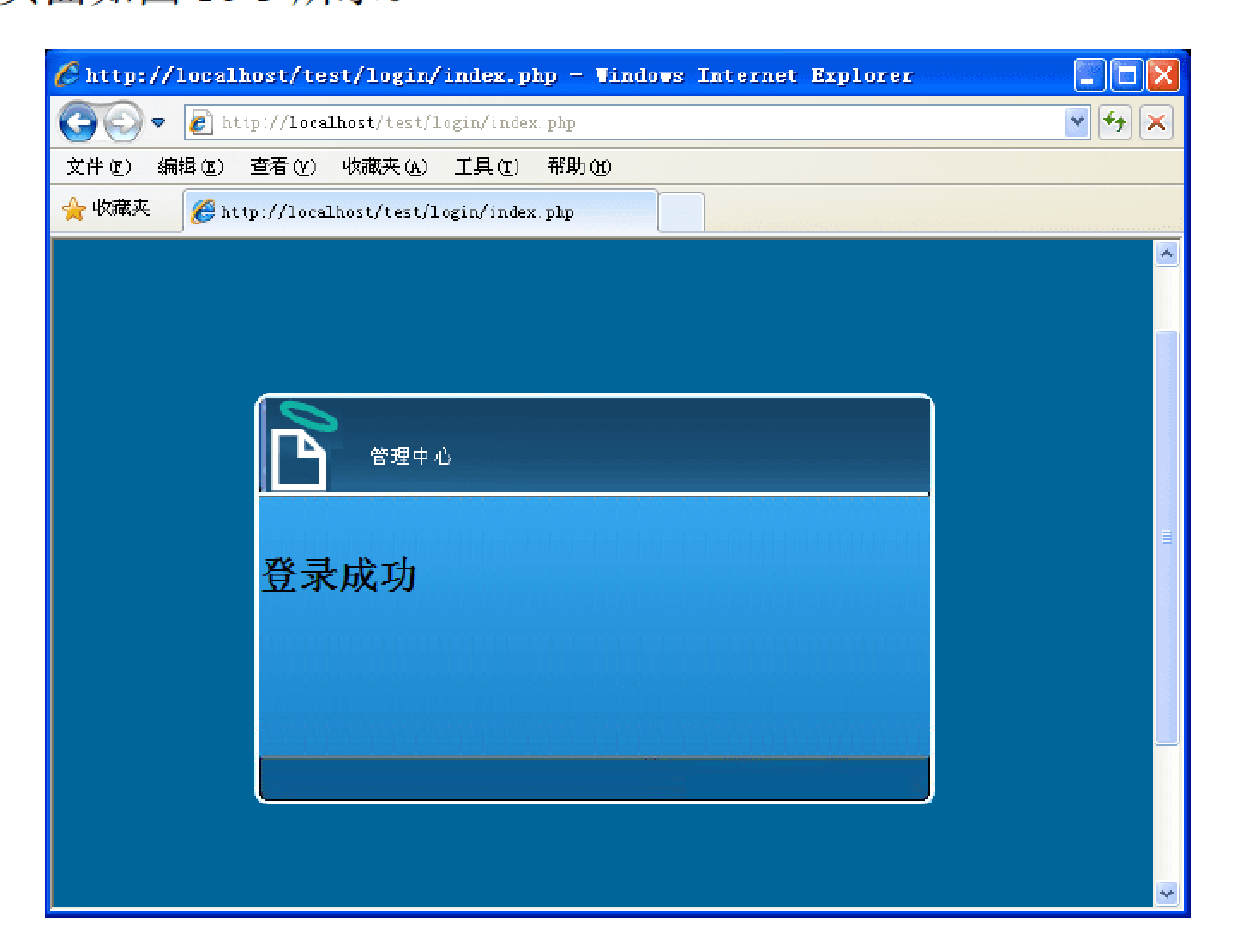


图 10-3　跳转页面

10.3.5　实例分析



源码解析：

上述实例中我们使用了在页面中对数据进行初步加密的方法，使用户在输入密码等重要信息时，不至于被别人看到。采取这种方式的具体方法是将密码输入框的 type 属性改为 password。

10.4　如何使用 PHP 实现加密

在 PHP 中，加密技术主要用来加密信息、产生校验和摘要。使用加密技术可大大地增强安全性能。PHP 对于一些加密算法提供了非常有利地支持。这一节里，我们对常用的 PHP 内置的加密函数进行介绍。

 **视频教学：**光盘\videos\10\PHP 内置加密函数.avi　　长度：8 分钟

10.4.1　基础知识——PHP 内置加密函数

在 PHP 中常用的加密函数有 md5()、crypt()和 sha1()，在这里我们将对这些函数进行介绍。

1．md5()函数

md5()函数是使用 MD5 信息摘要算法(MD5 Message-Digest algorithm)计算出输入字符串的

md5 散列值，语法如下所示：

```
string md5(string $str [, bool $raw_output ])
```

在函数中，$str 是需要加密的字符串，$raw_output 是函数中预设的一个值，默认为 false 时，函数会以一个 32 位字节的十六进制数据传送散列值，如果值为 true，则函数会以一个 16 位字节长度的原始二进制数据传送散列值。

md5()函数是一种单向散列算法，我们没有办法将使用 md5()函数的数据进行逆散列，所以还可以作为密码验证。如以下例子代码：

```
$password = "624879513";
$mpass = md5($passwor);
echo "输出结果："．$mpass;//输出结果：76a7d786dd3507497f78f6cb34daed2e
```

在这个小例子中，我们可以看到.$mpass 的输出结果是一系列的散列值。这样我们就可将这些加密的散列值存储到数据库中，以提高数据的安全性。

2．crypt()函数

crypt()函数也是一个单向的加密函数，在使用时，我们可以把一个输入的密码和一个预存的密码进行比较，无需任何的解密过程。函数语法如下所示：

```
string crypt (string $str [, string $salt ])
```

该函数采用以 UNIX 的 DES 为基础的加密算法(或者操作系统提供的替代算法)来加密输入的字串，变量$str 就是被加密的字符串，salt 参数为加密时使用的干扰串，如果省略，会随机生成一个干扰串。如下例子：

```
$password = "624879513";
$cpass = crypt($password,"hhhh");
echo "输出结果："．$cpass;//输出结果：hh125ym72KVzQ
```

crypt()函数将一串明文作为它的第一个参数字，对它应用 salt 会影响加密算法的随机性，并生成输入明文的单向暗文。如果不提供 salt，则 PHP 通常默认其系统 salt，它的长度和值如表 10-6 所示。

表 10-6　算法 salt 的取值和长度

取　值	长　度	详细信息
CRYPT_STD_DES	2 个字符	如果支持 2 字符 salt 的 DES 加密，则设置为 1，否则为 0
CRYPT_EXT_DES	9 个字符	如果支持 9 字符 salt 的 DES 加密，则设置为 1，否则为 0
CRYPT_MD5	12 个字符	如果支持以1开始的 12 字符 salt 的 MD5 加密，则设置为 1，否则为 0
CRYPT_BLOWFISH	16 个字符	如果支持以2或$2a$开始的 16 字符 salt 的 Blowfish 加密，则设置为 1，否则为 0

3．sha1()函数

Shal()函数是用来计算一个字符串的 SHA-1 散列值，其语法格式如下：

```
string sha1 (string $str [, bool $raw_output ])
```

该函数以字符串的方式传送输入字符串的 SHA-1 散列值，其中参数$str 为输入的字符串，$raw_output 是函数预设的一个参数，参数值为真时，函数会传送一个长度为 20 个字节的二进制数据；参数为假时，会传送一个长度为 40 个字节的十六进制数据。

下面我们看一个例子，代码如下：

```
$spass = sha1($password,false);
echo "输出结果：".$spass."<br>";
//输出结果：0bbd9091033c6d889e8bc29b16eab0a9d19fc96d
```

我们可以看到，当参数设为 false 时，使用 shal 函数对字符串计算出的值是一个长度为 40 个字节的十六进制数据。

10.4.2　实例描述

在 PHP 的 Web 开发中，虽然我们可以在前台页面的输入框中使用 PASSWORD 属性，使用户在输入时看不到输入的内容。但是，这些信息无论是存储到数据库还是在网络之间传送，都是以明文的形式传送的。对于一些攻击者来说，这些信息就是透明的了。因此，这里的实例我们将使用 PHP 的加密函数对数据进行加密后，再进行数据存储和传送操作。

10.4.3　实例应用

【例 10-4】 使用 md5()函数加密密码

(1) 新建一个名为 11.4.htm 的前台登录页面，在页面中我们设计一个登录表单，用户可以输入用户名和密码。其部分代码如下所示：

```
<TABLE class=tableborder cellSpacing=0 cellPadding=0 width=427
     align=center border=0>
       <TBODY>
       <TR>
         <TD background=11.3_files/index_hz01.gif colSpan=3
           height=64>  </TD></TR>
       <TR>
         <TD style="FONT-SIZE: 1px; LINE-HEIGHT: 1px" width=3
         background=11.3_files/index_hz02.gif></TD>
         <TD style="BORDER-TOP: #666666 1px double" vAlign=top
         background=11.3_files/index_hz03.gif height=162><BR><BR><!--
-->
           <TABLE id=adminlogin cellSpacing=0 cellPadding=0 width="100%"
           align=center border=0>
             <FORM name=form1 action=11.4.php method=post>
             <TBODY>
             <TR>
               <TD class=td align=right width="33%">账 号:</TD>
               <TD class=td width="67%"><INPUT
```

```
                style="BORDER-RIGHT: #0d5c95 1px solid; BORDER-TOP: #0d5c95 1px
solid; BACKGROUND: #fff; BORDER-LEFT: #0d5c95 1px solid; WIDTH: 140px;
BORDER-BOTTOM: #0d5c95 1px solid; HEIGHT: 21px"
                name=loginname> </TD></TR>
            <TR>
              <TD class=td align=right width="33%">密 码:</TD>
              <TD class=td width="67%"><INPUT
                style="BORDER-RIGHT: #0d5c95 1px solid; BORDER-TOP: #0d5c95 1px
solid; BACKGROUND: #fff; BORDER-LEFT: #0d5c95 1px solid; WIDTH: 140px;
BORDER-BOTTOM: #0d5c95 1px solid; HEIGHT: 21px"
                type=password size=21 name=loginpwd> </TD></TR><!---->
            <TR>
              <TD class=td width="33%"> </TD>
              <TD class=td width="67%"><INPUT type=image height=33 width=83
                src="11.3_files/index_hz04.gif" name=Submit>
            </TD></TR></FORM></TBODY></TABLE><!----></TD>
        <TD style="FONT-SIZE: 1px; LINE-HEIGHT: 1px" width=3
        background=11.3_files/index_hz02.gif></TD></TR>
      <TR>
        <TD background=11.3_files/index_hz05.gif colSpan=3
          height=127> 
```

(2) 新建一个名为 11.4.php 的文件，在文件中添加连接数据库代码，如下所示：

```
<?php
session_start();
$conn= mysqli_connect("localhost","root","root","userman");
$conn->query('set names gb2312');
if(empty($conn))
    {
      die("数据库连接失败");
    }
```

(3) 在 11.4.php 文件中，我们要对从前台得到的数据进行加密处理，代码如下：

```
$logname = $_POST['loginname'];
$logpass = $_POST['loginpwd'];
$logpassmd = md5($logpass);
```

(4) 对登录进行密码验证，其部分代码如下：

```
$str="select * from admin";
        $result1=$conn->query($str);
        $row=$result1->fetch_row();
if($row[2]!=$logpassmd){
?>

<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
  <TBODY>
  <TR>
    <TD height=600><BR><BR><BR><BR>
```

```
    <TABLE class=tableborder cellSpacing=0 cellPadding=0 width=427
    align=center border=0>
      <TBODY>
      <TR>
        <TD background=11.3_files/index_hz01.gif colSpan=3
          height=64>  </TD></TR>
      <TR>
        <TD style="FONT-SIZE: 1px; LINE-HEIGHT: 1px" width=3
        background=11.3_files/index_hz02.gif></TD>
        <TD style="BORDER-TOP: #666666 1px double" vAlign=top
        background=11.3_files/index_hz03.gif height=162><BR><BR><!--
-->
          <TABLE id=adminlogin cellSpacing=0 cellPadding=0 width="100%"
          align=center border=0>
    <h2><?
      echo "密码不正确<br>";
         $temp="请重新登录，谢谢!";
         echo $temp;
         echo"<a href=11.4.htm>返回</a>";
   }else{?> </h2>
       <h2><?echo "登录成功";}?></h2>
          </TABLE><!----></TD>
```

(5) 保存文件。

10.4.4 运行结果

将 11.4.htm 文件在服务器中运行，我们可以看到如图 10-4 所示的结果。

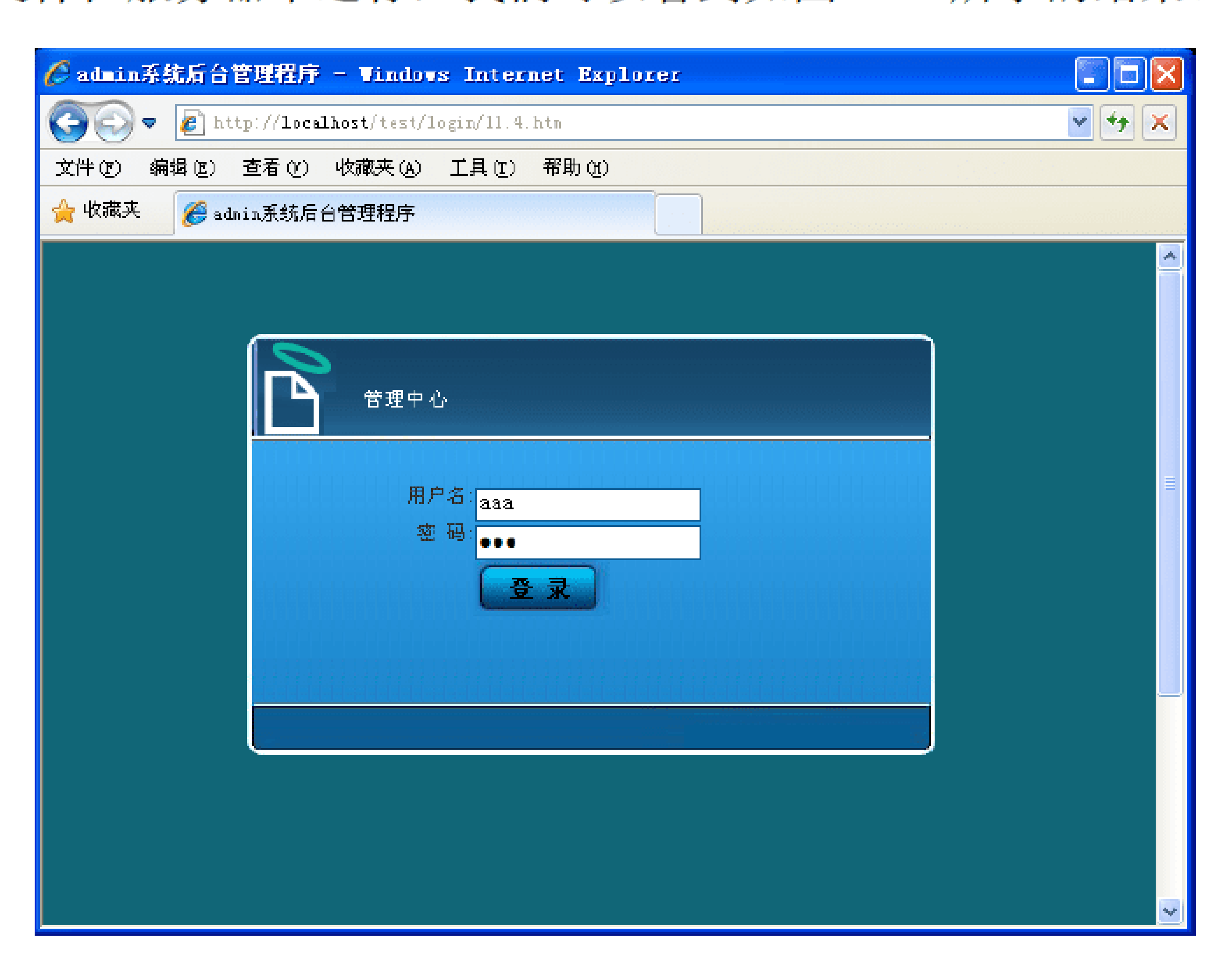


图 10-4 用户登录

当用户填写完用户名和密码后，单击【登录】按钮，如果用户输入结果和 11.4.php 文件中

加密的数据也就是数据库中的数据不一样，则会显示如图 10-5 的结果。

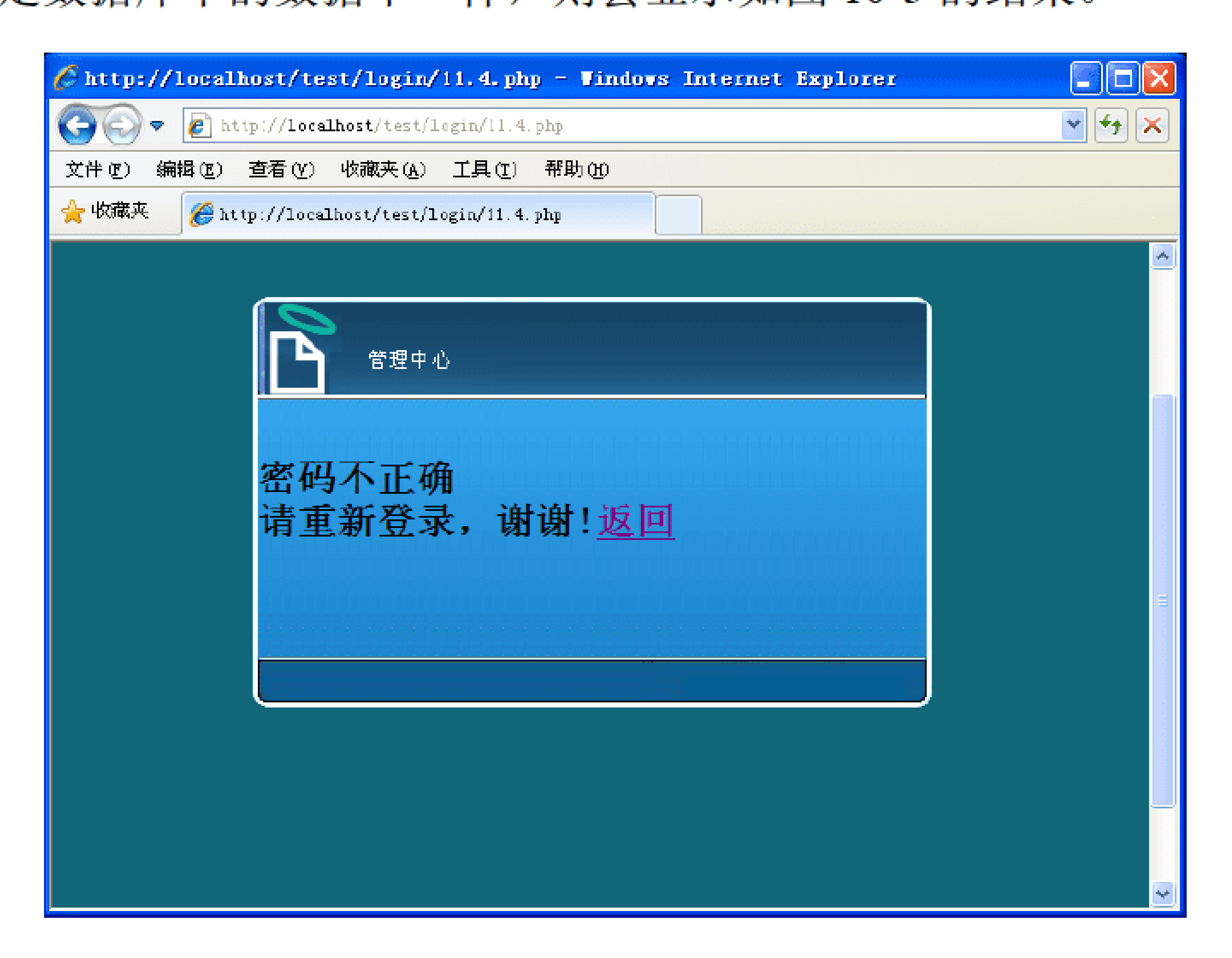


图 10-5　重新登录

10.4.5　实例分析



源码解析：

上述实例中我们采用了对用户从页面输入的信息进行了 md5 加密，然后把加密后的数据存入数据库，当用户输入用户名和密码时，逻辑处理页面就会对用户输入的数据进行加密后再与数据库中的加密数据进行比较。这样处理数据使得页面中和数据库中的数据都是加密后的密文，这样对于攻击者来说很难找到想要的明文数据。

10.5　强大的 PHP 加密扩展库

PHP 中除了内置的加密函数外，还支持了非常强大的 PHP 外部加密函数扩展包。使用这些扩展包，可以使我们的数据更加安全。

**视频教学：**光盘\videos\10\PHP 的加密扩展库.avi　　　　长度：12 分钟

10.5.1　基础知识——PHP 加密扩展库

PHP 的加密扩展库有很多，有提供双向加密的，还有为大量散列算法提供接口的，如表 10-7 所示：

表 10-7　PHP 加密扩展库

扩展包和函数库	详细描述
Mcrypt	提供广泛的加密功能，可以双向加密，用来加密大型档案或数据流
Mhash	支持最流行的散列算法，如 MD5
Crypt_Blowfish	可以进行快速双向加密，可以选择使用或不使用密钥
Crypt_RSA	提供 RSA 一样的密钥生成、加密/解密、数位签署及签署验证功能
Crypt_HMAC	这个类可用来计算兼容 RFC 2104 的散列值
Crypt_DiffieHellman	这是一个在 PHP5 上的 Diffie-Hellman 密钥交换协议

PHP 提供了众多的加密扩展，但我们比较常用的只有两三种。下面我们在这一节里将对 PHP 常用的库进行详细讲解。

1. Mcrypt

Mcrypt 中有大量的加密函数文件和数据流进行加密，Mcrypt 的加密都是针对固定长度的数据块，一般为 64 或 128 个字节。但由于明文的长度是不固定的，再加上相同的密钥来加密相同的明文，所示我们会得到相同的结果。Mcrypt 支持多种区块的加密算法包括 Blowfish、DES、TripleDES、SAFER- SK128、TWOFISH、TEA、RC2、3-WAY、SAFER-SK64 6 种加密模式，其中 6 种模式如表 10-8 所示。

表 10-8　加密模式

模式缩写	使用例子	详细信息
CBC	MCRYPT_MODE_CBC	密文块链模式，用来加密文件
CFB	MCRYPT_MODE_CFB	密文反馈模式，建议用来加密字节流
STREAM	MCRYPT_MODE_ STREAM	特色的流模式，需要资料流算法时使用，例如 WAKE 或 RC4
ECB	MCRYPT_MODE_ECB	电子密码本模式，适合随机数据，你可以用这种模式来加密不同的密码
OFB	MCRYPT_MODE_OFB	8 位输出反馈模式，专门用在不容许出错的应用系统
NOFB	MCRYPT_MODE_NOFB	N 位输出反馈模式，兼容 OFB，但更加安全

通过 Mcrypt，PHP 的加密扩展可以使用 30 多个函数，我们在这里只介绍最重要的两个函数，加密函数 mcrypt_encrypt()和解密函数 mcrypt_decrypt()。

1)　mcrypt_encrypt()函数

mcrypt_encrypt()是 Mcrypt 数据加密包中的一个加密函数。其语法如下：

```
string mcrypt_encrypt(string dipher,string key ,string data,string mode[,string iv]);
```

在这个函数中，参数 dipher 是在函数中指定的加密算法，参数 key 是确定使用的加密密钥，

参数 mode 是指定的 6 个加密模式中的一个加密模式，参数 data 是要加密的数据，返回加密后的结果，而参数 vi 是对 cbc、cfb、ofb 以及流模式中使用的某种算法。如以下例子我们对一段明文进行加密：

```
$ivs = mcrypt_get_iv_size(MCRYPT_DES,MCRYPT_MODE_CBC);
$iv = mcrypt_create_iv($ivs,MCRYPT_RAND);
$key = "TMFNHFK";
$mess = "我想加密这条数据";
$nec = mcrypt_encrypt(MCRYPT_DES,$key,$mess,MCRYPT_MODE_CBC,$iv);
echo bin2hex($nec);
```

在这个例子中我们采用了 DES 的加密模式对数据进行加密，其代码运行结果如下：

```
24791fa416cec526dc0759d5a5a2ab0f
```

2)　mcrypt_decrypt()函数

mcrypt_decrypt()函数是进行解密的函数，用于解密 mcrypt_encrypt()函数中的 cipher。其语法如下：

```
string  mcrypt_decrypt(string  dipher,string  key  ,string  data,string
mode[,string iv]);
```

在加密函数的例子中，我们加入以下代码就可以对已经加密的数据进行解密。

```
echo mcrypt_decrypt(MCRYPT_DES,$key,$nec,MCRYPT_MODE_CBC,$iv);
```

2. Mhash

Mhash 是一个免费的函数开源库，它为 PHP 加密的大量散列值算法提供了一个有用的接口。我们可以用这些算法计算校验信息和进行数据签证等，散列还可以用于处理密码。

PHP 得到 Mhash 的扩展之后，我们可以用到它的 5 个函数，在这里我们只讲一个最重要的 Mhash()函数。其语法格式如下：

```
string mhash(int hash,string data[,string key]);
```

在函数中，参数 hash 是设置散列算法，参数 data 是函数要进行散列的密码。如以下例子中我们可以看到要散列密码的散列值：

```
$PASS = "PINGNIANGAN";
  $PASSMHASH = mhash(MHASH_MD5,$PASS);
  echo "该密码的散列值是：".bin2hex($PASSMHASH);
```

这段代码会返回以下结果：

```
该密码的散列值是：2a95f3d7706cbeafc8f745e07d53a084
```

这样我们就可以计算出密码的散列值了，然后将密码的散列值保存到数据库中，就可以增加密码的安全性。在这我们采用了 MHASH_MD5 的散列算法，目前 Mhash 函数支持的散列算法有很多，如表 10-9 所示。

表 10-9　Mhash 支持的散列算法

散列算法			
MHASH_ADLER32	MHASH_HAVAL128	MHASH_MD4	MHASH_SHA256
MHASH_CRC32	MHASH_HAVAL160	MHASH_MD5	MHASH_TIGER
MHASH_CRC32B	MHASH_HAVAL192	MHASH_RIPEMD160	MHASH_TIGER128
MHASH_GOST	MHASH_HAVAL256	MHASH_SHA1	MHASH_TIGER160

3. PEAR 的 Crypt_RSA PEAR 套件

这种套件能够让你用任意长度的密钥来加密数据，它是以 RSA 区块加密技术为基础，支持双向加密，它支持任意长度的密钥来加密和解密。Crypt_RSA 需要执行密集的数学计算，所以我们需要表 10-10 中的任意一个模块才能使用。

表 10-10　密集计算扩展模块

模块名称	支持版本
PECL big_int 扩展模块	需要 PEAR 包 1.0.3 以上版本
PHP GMP 扩展模块	PHP 版本
PHP BCMath 扩展模块	PHP4/PHP5

至于 PEAR 包的安装在这里不再讲解。安装好后，我们就可以使用了，如以下代码对数据进行的加密：

```
<?php
  require_once 'Crypt/RSA.php';
   // 产生一双对称密钥
   function generate_key_pair()
   {
      global $public_key,$private_key;
      $key_pair = new Crypt_RSA_KeyPair(32);
      // 从这双密钥中提取公钥
      $public_key = $key_pair->getPublicKey();
      // 从这双密钥中提取密钥
      $private_key = $key_pair->getPrivateKey();
   }
   $file = 'textfile.txt';
   generate_key_pair();
   $plain_text = file_get_contents($file);
   // 把公钥表达为一个字符串
   $key = Crypt_RSA_Key::fromString($public_key->toString());
   $rsa_obj = new Crypt_RSA;
   // 用密钥 $key 加密 $plain_text
   $encrypted = $rsa_obj->encrypt($plain_text, $key);
   $encrypted_file = @fopen('encrypted.txt','w');
   $ok_encrypt = fwrite($encrypted_file,$encrypted);?>
```

4. Crypt_HMAC 来产生散列值

PEAR 的 Crypt_HMAC 套件有一个类可以用来计算 RFC2104 兼容的散列值，使用方法很简单，你只需告诉它你的密钥、散列计算方法和明文就可以了，它还支持 MD5 和 SHA-1 算法。语法如下所示：

```
$var = new Crypt_HMAC(string key, string dipher);
```

其中 key 是要提供的密钥，而 dipher 是对数据应用的散列算法。然后我们通过 Crypt_HMACCI 函数就可以计算出明文的散列值。如以下例子：

```
<?php
  require_once 'HMAC.php';
  // 把字符"defre" 重复 40 次来产生一个密钥
  $key = str_repeat(chr(defre), 40);
  // 产生一个 Crypt_HMAC 类的实体
  $crypt = new Hmac($key, 'md5');
  // 散列函数
  echo $crypt->sha1('Hello');
  $key = str_repeat(chr(0xcc), 10);
  $data = str_repeat(chr(0xdd), 50);
  // 把散列函数的密钥设定为 $key
  $crypt->setKey($key);
  echo $crypt->sha1($data)."\n";
?>
```

10.5.2 实例描述

仅用 PHP 内置的加密函数对信息进行加密，其数据的安全性还是很低的，这样就需要我们利用 PHP 强大的加密扩展库对数据进行更安全的加密。下面就让我们来看一个示例吧。

10.5.3 实例应用

【例 10-5】 动态加密密码

(1) 新建一个名为 11.5.htm 的文件，在这个文件中我们要创建一个登录界面，用户可以输入用户名和密码，对于输入的密码我们将在逻辑处理部分进行动态加密并进行校验。部分表单代码如下：

```
 <FORM id=Login name=Login action=login.php method=post>

    <TD class=form_item_title><LABEL
      id=Login_userName_title>用户名<SPAN
      class=notempty>*</SPAN></LABEL></TD>
    <TD class=form_item_input><INPUT id=userName size=10
      name=userName></TD></TR>
  <TR>
```

```
        <TD> </TD>
        <TD>
          <DIV id=Login_userName_status></DIV></TD></TR>
      <TR>
        <TD class=form_item_title><LABEL
          id=Login_password_title>密码<SPAN
          class=notempty>*</SPAN></LABEL></TD>
        <TD class=form_item_input><INPUT id=password
          type=password size=10 name=password></TD></TR>
      <TR>
        <TD> </TD>
        <TD>
          <DIV id=Login_password_status></DIV></TD></TR>
      <TR>
        <TD> </TD>
        <TD class=td width="67%"><INPUT type=image height=33 width=83
    src="11.5.files/index_hz04.gif" name=Submit>
</TD>
        <TD> </TD></TR></TBODY></TABLE></FORM>
```

(2) 新建一个逻辑处理文件，命名为 login1.php，在页面中，先于数据库建立连接，其代码如下：

```
session_start();
$conn= mysqli_connect("localhost","root","root","userman");
$conn->query('set names gb2312');
if(empty($conn))
    {
      die("数据库连接失败");
    }
```

(3) 取出表单数据，并使用 mcrypt_encrypt()函数对从表单中取出的密码进行加密，其代码如下：

```
$logname = $_POST['userName'];
$logpass = $_POST['password'];
$ivs = mcrypt_get_iv_size(MCRYPT_DES,MCRYPT_MODE_CBC);
$iv = mcrypt_create_iv($ivs,MCRYPT_RAND);
$key = "TMFNHFK";
$necpass = mcrypt_encrypt(MCRYPT_DES,$key,$logpass,MCRYPT_MODE_CBC,$iv);
```

(4) 取出数据库中的密码，对其进行加密，其代码如下所示：

```
$str="select * from admin where name='".$logname."'";
      $result1=$conn->query($str);
      $row=$result1->fetch_row();
$necrow = mcrypt_encrypt(MCRYPT_DES,$key,$row[2],MCRYPT_MODE_CBC,$iv);
```

(5) 对用户输入密码进行校验，代码如下所示：

```
if($necrow!=$necpass){?>
 <?echo "密码不正确<br>";
          $temp="请重新登录，谢谢!";
          echo $temp;?>
         <? echo"<a href=11.4.htm>返回</a>";
    }else{?>
        <?echo "登录成功";}?>
```

(6) 保存两个文件。

10.5.4　运行结果

将 11.5.htm 文件在浏览器中运行，我们可以看到如图 10-6 所示的用户登录页面。



图 10-6　用户登录界面

在该界面中输入相应的用户名和密码，单击【登录】按钮，就会跳转到逻辑处理页面 login1.php。如果用户输入的密码进行加密后的密码和从数据库中取出的加密后的密码相同，用户就登录成功，如图 10-7 所示，否则登录失败，如图 10-8 所示。



图 10-7　登录成功



图 10-8　登录失败

10.5.5 实例分析



源码解析：

上述实例中我们使用了 Mcrypt 中 mcrypt_encrypt()函数对用户输入的密码和从数据库中取出的密码进行了加密，并通过加密后的值进行了用户的登录校验。

10.6 常见问题解答

10.6.1 用 PHP 写会员注册密码加密问题



用 PHP 写会员注册密码加密问题。

网络课堂：http://bbs.itzcn.com/thread-13622-1-1.html。

大家看看这个注册页面 register.php 代码：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>无标题文档</title>
</head>

<body><br /><br />
<form name="form1" action="lr.php" method="post">
 <input type="text" name="user" maxlength="20" size="20" /><br /><br />
 <input type="password" name="pwd" maxlength="20" size="20" /><br /><br />
 <input type="submit" name="submit" value="login" />
</form>
</body>
</html>
```

是不是要在此页面进行 md5 加密再传到 lr.php 啊，但是那个 pwd 还是有可能被截获。具体该怎么做?请高手指教。

【解决办法】：数据被截获是无法避免的，除非用 SSL 加密。

比较安全的方法有以下几种。

(1) 数据正常提交至 lr.php。

(2) 获取一个当前时间 time()。

(3) 将密码组合 time()后 md5。

(4) 将组合后的密码 md5 值和之前获取的 time()一起存入数据库。

当今后需要登录时首先获取用户名对应的 time()，再组合上密码进行 md5，和数据库内的

密码 md5 对比。

10.6.2　PHP 3DES 加密问题



PHP 3DES 加密问题。
网络课堂：http://bbs.itzcn.com/thread-13623-1-1.html。

大家帮我看看下面的内容：
3des 加密信息：
Key：E0JW20IECVC21Y79Q6J6ORHX
原文：fsti
加密结果：3F3756AD34C75CF8
用 PHP 怎么实现加密后同样结果？希望贴个源码出来，非常感谢。
【解决办法】：3des 应该是通过 DES 加密的方式去加密的，如以下两段代码。

```
<?php
   $key = "this is a secret key";
   $input = "Let us meet at 9 o'clock at the secret place.";

   $td = mcrypt_module_open('tripledes', '', 'ecb', '');
   $iv = mcrypt_create_iv (mcrypt_enc_get_iv_size($td), MCRYPT_RAND);
   mcrypt_generic_init($td, $key, $iv);
   $encrypted_data = mcrypt_generic($td, $input);
   mcrypt_generic_deinit($td);
   mcrypt_module_close($td);
?>
<?php
//.h
function mdecrypt_3des($input,$key){
 $td = mcrypt_module_open(MCRYPT_3DES,'',MCRYPT_MODE_ECB,'');
 $blocksize = mcrypt_enc_get_block_size($td);
 $keysize = mcrypt_enc_get_key_size($td);
 $iv_size = mcrypt_enc_get_iv_size($td);
 $iv = mcrypt_create_iv ($iv_size,MCRYPT_RAND);
 $input_len = strlen($input);
 $padsize = $blocksize-($input_len%$blocksize);
 $input .= str_repeat(pack ('C*',$padsize),$padsize);
 @mcrypt_generic_init($td, $key, $iv);//key 最长允许 24 位 ?

 //return urlencode(base64_encode(mcrypt_generic($td,$input)));//加密
 return
rtrim(mdecrypt_generic($td,base64_decode(urldecode($input))),"\u0008");//解密
}
?>
```

10.6.3　打开服务器 PHP 安全模式 PHP 运行变慢问题



打开服务器 PHP 安全模式 PHP 运行变慢问题。
网络课堂：http://bbs.itzcn.com/thread-13624-1-1.html。

为什么打开服务器 PHP 安全模式 php 运行变慢了？程序运行速度由 0.03(s)上升到了 0.3(s)，服务器组件: IIS6+PHP5+MySQL5+Zend3.3.0a+eAccelerator v0.9.5.2。请各位给点建议，谢谢！

【解决办法】：你是说开启 safe_mode 吗？限制多了，速度自然慢，可以关闭的，影响不大。下面列出几个影响速度的原因。以下设置不影响速度的，可以考虑。

1. 关闭危险函数

如果打开了安全模式，那么函数禁止是可以不需要的，但是我们为了安全还是考虑进去。比如，我们觉得不希望执行包括 system()等在内的能够执行命令的 php 函数，或者能够查看 php 信息的 phpinfo()等函数，那么我们就可以禁止它们。

```
disable_functions = system,passthru,exec,shell_exec,popen,phpinfo
```

如果你要禁止任何文件和目录的操作，那么可以关闭很多文件操作。

```
disable_functions=chdir,chroot,dir,getcwd,opendir,readdir,scandir,fopen,
unlink,delete,copy,mkdir,rmdir,rename,file,file_get_contents,fputs,fwrite,
chgrp,chmod,chown
```

以上只是列了部分比较常用的文件处理函数，你也可以把上面执行命令函数和这个函数结合，就能够抵制大部分的 phpshell 了。

2. 关闭 PHP 版本信息在 http 头中的泄漏

我们为了防止黑客获取服务器中 php 版本的信息，可以关闭该信息泄漏在 http 头中。

```
expose_php = Off
```

3. 关闭注册全局变量

在 PHP 中提交的变量，包括使用 POST 或者 GET 提交的变量，都将自动注册为全局变量，能够直接访问，这是对服务器非常不安全的，所以我们不能让它注册为全局变量，把注册全局变量选项关闭。

```
register_globals = Off
```

当然，如果这样设置了，那么获取对应变量的时候就要采用合理方式，比如获取 GET 提交的变量 var，那么就要用$_GET['var']来进行获取，这个问题 php 程序员要注意。

10.7　习　　题

一、填空题

(1) ________指令能开启 PHP 的安全模式。

(2) ________的作用是禁用你希望禁用的函数名列表。

(3) ________指令用于设置一个路径，指定提供 PHP 文件的根目录。

(4) ________指令用于创建一个基目录，将限制所用的文件只能在这个文件中进行。

(5) ________指令指定用户主目录的一个用户名，规定 PHP 脚本必须在这里才能执行。

(6) ________指令主要负责 Apache 的服务器的版本、服务器名、端口和编译模块有关的一行输出。

(7) 加密技术分为两个方面，一个是对称加密，另一个是________。

(8) 非对称加密算法需要两个密钥：公开密钥(publickey)和________。

(9) Shal()函数是用来计算一个字符串的________散列值。

(10) Mcrypt 的加密都是针对固定长度的数据块，一般为 64 或________个字节。

二、选择题

(1) 下面哪种不是 Mcrypt 中支持的加密模式？________

A. CBC　　B. STREAM　　C. OFB　　D. NOHB

(2) 下面哪种是 Mcrypt 中的解密函数？________

A. md5();　　B. mcrypt_encrypt();

C. mcrypt_decrypt();　　D. sha1();

(3) 以下不是 Mhash 函数支持的散列算法的是？________

A. MHASH_MD4　　B. MHASH_MD5

C. MHASH_RIPEMD160　　D. MHASH_SJI

三、上机练习

上机练习： 自己动手制作登录和注册页面，要求对用户输入的信息进行加密。

练习要求：首先制作用于用户注册信息和登录的页面，在进行后台处理时使用 MD5 和 Mcrypt 对用户的密码和用户信息进行加密。执行效果如图 10-9 和图 10-10 所示。

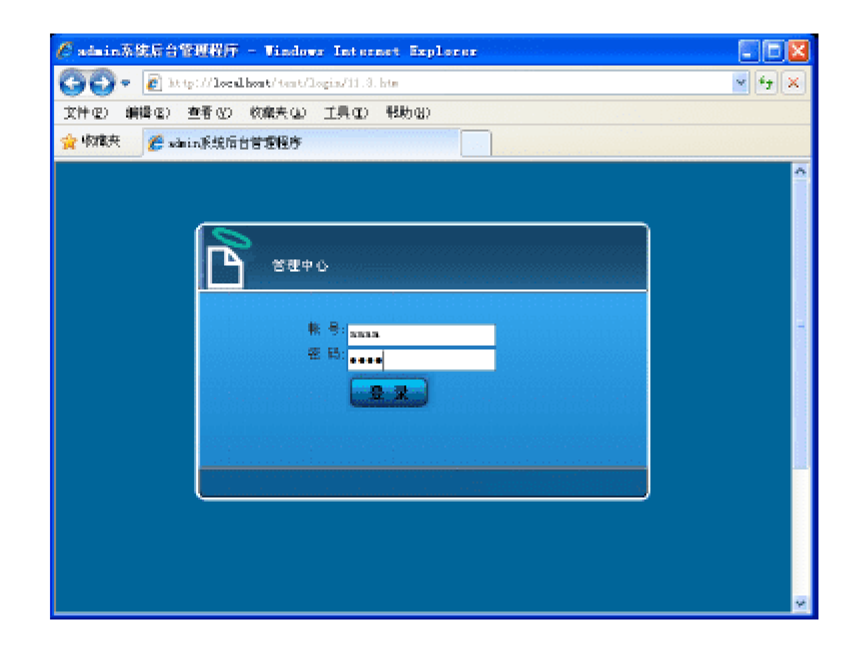


图 10-9　登录页面

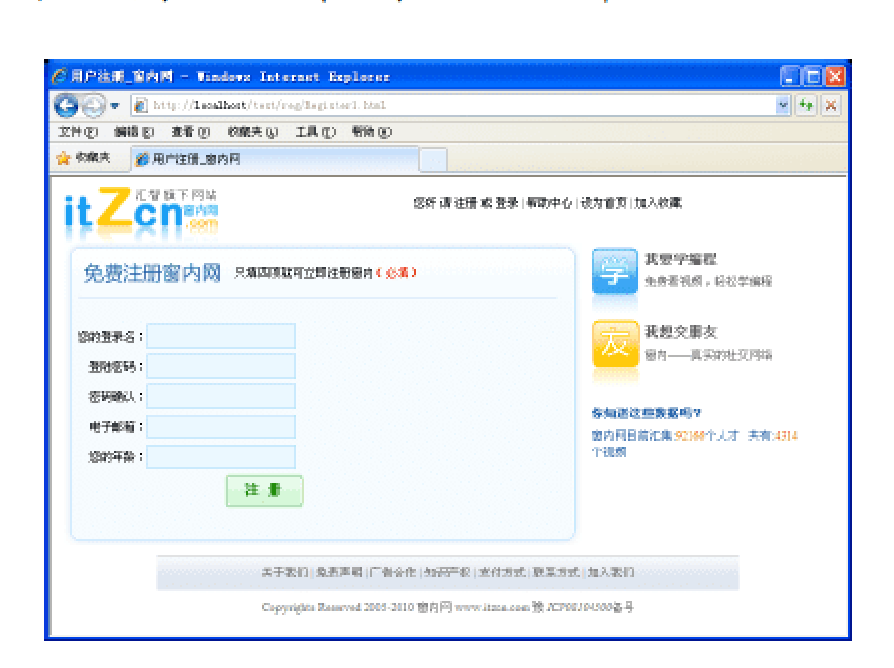


图 10-10　注册页面



第 11 章　套用装饰好的页面

内容摘要：

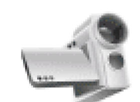
在之前使用 PHP 做 Web 开发时，可能会有各种问题，虽然没有尽善尽美的解决方案，但已经有很多方案可以将网站的表现与逻辑几乎完全分离，这些解决方案称为模板化引擎(templating engine)，它们正在逐步消除由于缺乏层次分离而带来的难题，本章将介绍模版化引擎在 PHP 中的应用，并且特别强调目前流行的 PHP 模板化解决方案 Smarty。

学习目标：

- 了解 Smarty 简介。
- 掌握 Smarty 安装配置。
- 练习使用 Smarty 模板。
- 熟知 Smarty 模板的变量修饰符。
- 熟悉 Smarty 模板的流程结构和语句。
- 掌握如何创建 Smarty 配置文件。

11.1 Smarty 简介

Smarty 是一个基于 PHP 开发的 PHP 模板引擎。它提供了逻辑与外在内容的分离，简单来讲，目的就是要使 PHP 程序员同美工分离，使程序员改变程序的逻辑内容不会影响到美工的页面设计，美工重新修改页面不会影响到程序的逻辑，这在多人合作的项目中显得尤为重要。

 **视频教学**：光盘\videos\11\Smarty 简介.avi　　长度：6 分钟

11.1.1 Smarty 的优点

1．开发速度

Smarty 编写的程序速度可以获得最大的提高，这一点是相对于其他的模板引擎技术而言的。

2．编译型

Smarty 编写的程序在运行时要编译成一个非模板技术的 PHP 文件，这个文件采用了 PHP 与 HTML 混合的方式，在下一次访问该模板时将 WEB 请求直接转换到这个文件中，而不再进行模板重新编译，必须是在源程序没有改动的情况下。

3．缓存技术

Smarty 选用的一种缓存技术，它可以将用户最终看到的 HTML 文件缓存成一个静态的 HTML 页面，当设定 Smarty 的 cache 属性为 true 时，在 Smarty 设定的 cachetime 期内将用户的 WEB 请求直接转换到这个静态的 HTML 文件中来，这相当于调用一个静态的 HTML 文件。

4．插件技术

Smarty 可以自定义插件。插件实际就是一些自定义的函数。

11.1.2 Smarty 流程

通过图 11-1 的 Smarty 流程图，可以清楚地观察到 Smarty 在 PHP 中的工作原理和步骤。

和其他 PHP 模板引擎不同的地方在于，Smarty 最大的特点是“模板编译”。从流程图上可以看到，对于一个模板，Smarty 会将它编译为 PHP 文件，最后输出的时候执行的是编译过的 PHP 程序，这样做比普通的模板变量正则替换在效率方面有了很大的提高。

获取模板文件

模板是否需要重新编译

是

将模板编译为 PHP文件

否

获取模板具体变量值

运行模板编译后的
PHP文件，输出结果

结束

图 11-1　Smarty 流程图

11.2　用户登录页面

通过简单地对 Smarty 模板进行介绍，大家对该模板一定有了进一步的了解，下面是对该模板的安装配置进行讲解，我们可以想象 Smarty 应该有怎样的最后表现。我们知道 PHP 代码如果没有了模板解析的开销将有多快，我们也知道从一般的美工看来 PHP 语言是多么的“恐怖”，然而这一切可以被一种更简单的模板语法掩饰掉。我们应该怎样把这两种方法的长处结合起来呢？

 **视频教学：**光盘\videos\11\安装使用 Smarty.avi　　　　长度：10 分钟

11.2.1　基础知识——安装使用 Smarty

1．安装 Smarty

安装配置 Smarty 超级简单。首先，需要到 http://www.smarty.net/下载最新的稳定版本。然后按照以下步骤进行安装配置即可。

(1) 下载 Smarty 包，可以从官方站点下载：http://www.smarty.net/download。

(2) 解压缩 Smarty 包并找到合适的位置存放。解压之后文件夹名可能是 Smarty.x.x.x，把它修改成 smarty(此处只为将来使用方便起见，并非必须)，然后将含有 Smarty 文件的文件夹复

制到某一个目录下(建议不要放在你的网站目录下，这样别人可以通过路径访问到这些文件，如果非要放在网站目录下，那最好目录的名字起的生僻一些。这些文件即使被别人访问到，也未必造成安全问题，但是一切都为了以防万一)。

(3) 找到你的 php.ini 配置文件(可以通过运行 phpinfo()查看此文件路径，但一般都放在系统目录下)修改 php.ini 的 include_path 选项，把 smarty 的库文件路径加上，比如：

```
include_path = ".;d:\smarty\libs"
```

当然，可能希望将这个路径附加到 include_path 中其他路径后面，因为可以将各种库以相似的方式集成到应用程序中。记住，在对 PHP 的配置文件做出更改之后，需要重新启动 Web 服务器。另外，要让应用程序可以应用 Smarty 库，还可以有其他的方法，比如，可以简单地提供类库的完全绝对路径。另一种用法是设置一个名为 SMARTY_DIR 的预定义常量，指向 Smarty 类库路径，然后类库名以常量开头。因此，即使某个配置需要修改 php.ini 文件，也需要知道这并不会阻止使用 Smarty 模板。



php.ini 中一共有两处 include_path，一处是 Unix 下使用的，一处是 windows 下使用的，要修改 windows 下使用的。

(4) 最后创建存储 Smarty 模板和配置文件的 4 个目录。

- templates：放置所有网站模板。
- configs：放置在特定网站中使用的所有特殊的 Smarty 配置文件。
- templates_c：放置 Smarty 编译的所有模板。
- cache：在启用缓存特性的情况下，放置 Smarty 缓存的所有模板。

虽然在默认情况下，Smarty 认为这些目录与实例化 Smarty 类的脚本位于相同的目录，但建议将这些目录放在 Web 服务器文档根目录之外的位置。可以分别使用 Smarty 的 $template_Dir、$compile_dir、$config_dir 和$cache_dir 修改默认行为。所以，可以将 smarty.php 文件设置为以下代码所示：

```
<?php
include("smarty/libs/Smarty.class.php");
    $smarty = new Smarty;
    $smarty->template_dir = 'templates';
    $smarty->config_dir = 'config';
    $smarty->cache_dir = 'smarty_cache';
    $smarty->compile_dir = 'smarty_templates_c';
    //上面四行为使用 Smarty 前的必要参数配置
    $smarty->caching=false;         //关闭缓存
    $smarty->left_delimiter="{";    //左边界
    $smarty->right_delimiter="}";   //右边界
?>
```

完成上述步骤之后，就可以使用 Smarty 了，为了让大家对这个模板化引擎更加有兴趣，下面从简单的一个实例开始，讨论更有兴趣、更有用的特性。当然讨论的时候还会通过一些代码进行详细的示范。

2．使用 Smarty

使用 Smarty 与使用其他任何类库一样。对于初学者，只需要在执行脚本中使 Smarty 类库可用，想实现这点非常容易，使用 require()语句即可，代码如下所示。

```
include("smarty/libs/Smarty.class.php");
```

紧接着就是实例化 Smarty 类：

```
$smarty=new Smarty;
```

11.2.2　实例描述

开始利用这个特性时所需要做的工作只有这么多。下面通过一个简单的用户登录页面进行详细讲解。在登录页面，需要通过使用 PHP 将网站公告和页面标题数据显示在静态登录页面并且绑定到指定的位置。

11.2.3　实例应用

【例 11-1】 用户登录

以下代码是定义一个 templates/login.html 文件的 Smarty 设计模板：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title> {$title} </title>
</head>
欢迎{$content}登录我们的系统！
<body>
</body>
</html>
```

还要注意的是 Smarty 希望这个模板位于 templates 目录内，除非通过$template_dir 修改了模板目录。

index.php 文件中代码是给出了相应的应用程序逻辑，将适当的变量值传入 Smarty 模板，代码如下所示：

```
<?php
include("smarty.php");
    $content="站长提醒，注意用户名密码外漏！";
    $title="后台登录系统";
    $smarty->assign('content',$content);
    $smarty->assign('title',$title);
    $smarty->display("templates/login.html");
?>
```

上述实例展示了 Smarty 能够完全分离 Web 应用程序逻辑层和表现层。但是，这只是 Smarty 全部特性的一点点而已。在进入其他主体之前，有必要正式介绍一下实例中用来获取和显示 Smarty 模板的 display()方法。

display()方法基于 Smarty 的脚本中都会用到这个方法，因为它负责获取和显示有 template 引用的模板。可选择的参数 cache_id 指定缓存标识符的名称，另一个参数 compile_id 在维护同一页面的多个缓存时使用。

11.2.4 运行结果

通过以上的努力，该实例编写大致已经完成，接下来将页面部署到服务器上运行，结果如图 11-2 所示。



图 11-2 用户登录

11.2.5 实例分析



源码解析：

在上述示例中给出了一个简单的设计模板，该模板中有两个变量$title 和$name。两个变量都放在大括号中，大括号是 Smarty 的默认定界符，这些定界符告诉 Smarty 要对定界符所包围的内容完成某些操作。在以下实例中，唯一的动作就是用通过应用逻辑传入的应用值代替变量。不过，很快就能看到，Smarty 还能够完成大量的其他任务。

11.3 网 站 简 介

在很多情况下，应用程序将大量的成本花费在维护上面。因此呢，大家都希望应用程序是易于修改的，而程序开发人员则迫切地希望使用框架来实现这些改变。模板技术提供了一种简

洁的方式来生成动态的页面，并将程序逻辑和视图之间分离开来，从而使程序开发人员只专注于编写底层代码，页面设计人员只专注于视图方面的设计，这种方式不仅提高了开发效率，还使得应用程序在长时间运行后依然具有很好的维护性。因此下面通过使用 Smarty 模板技术完成网站介绍。

视频教学：光盘\videos\11\网站简介.avi　　　长度：8 分钟

11.3.1　实例描述

该实例主要功能是对网站的性质和网站的功能进行介绍，在这里，通过使用简单的 Smarty 模板功能实现网站简介显示功能，而且该页面的信息数据是通过 PHP 后台页面将数据传递到模板页面中，从而完成网站简介显示的功能。

11.3.2　实例应用

【例 11-2】　网站首页

首先创建静态首页面 index.html 在 templates 目录下，代码如下所示：

```
<div id="templatemo_menu_wrapper">
    <div id="templatemo_menu">
        <ul>
            <li><a href="#" class="current">{$navigation1}</a></li>
            <li><a href="#">{$navigation2}</a></li>
            <li><a href="#" target="_parent">{$navigation3}</a></li>
            <li><a href="#" target="_blank">{$navigation4}</a></li>
            <li><a href="#" class="last">{$navigation5}</a></li>
        </ul>
    </div>
</div>
<div id="templatemo_content_inner_wrapper">
    <div id="templatemo_content_wrapper">
        <div class="section_w500">
            <h2>网站简介</h2>
            <p class="em_text">{$content}</p>
            </div>
    </div>
</div>
```

在代码中，设置了页面显示导航和网站简介信息，这些信息是通过变量绑定在页面内，然后设置 smarty.php 文件，该文件用来设置 Smarty 模板，实例化 Smarty，代码如下所示.

```
<?php
include("smarty/libs/Smarty.class.php");
    $smarty = new Smarty;
```

```
    $smarty->template_dir = 'templates';
    $smarty->config_dir = 'config';
    $smarty->cache_dir = 'smarty_cache';
    $smarty->compile_dir = 'smarty_templates_c';
    //上面四行为使用 Smarty 前的必要参数配置
    $smarty->caching=false;        //关闭缓存
    $smarty->left_delimiter="{";   //左边界
    $smarty->right_delimiter="}";  //右边界
?>
```

上述代码中，template_dir 是用来放置 Smarty 编译的所有模板的位置，而 caching 则是用来设置是否关闭缓存，false 为关闭缓存。最后设置 left_delimiter 和 right_delimiter 为边界结束符。

接下来创建 index.php 文件设置将适当的变量值传入 Smarty 模板，代码如下所示：

```
<?php
include("smarty.php");
    $content="郑州汇智计算机技术服务有限公司成立于 2002 年。致力于把基于先进信息技术的最佳管理与业务实践普及到客户的管理与业务创新活动中，全面提供企业管理 ERP 软件、服务与解决方案，是中原地区较早的管理软件、客户关系管理软件、小型企业管理软件以及网站建设服务提供商.汇智科技公司是北京新网郑州地区核心代理服务商，提供域名注册、服务器托管和虚拟主机服务?";
    $title="网站首页";
    $navigation1="窗内学院";
    $navigation2="窗内论坛";
    $navigation3="技术文档";
    $navigation4="娱乐空间";
    $navigation5="窗内商城";
    $smarty->assign('title',$title);
    $smarty->assign('content',$content);
    $smarty->assign('navigation1',$navigation1);
    $smarty->assign('navigation2',$navigation2);
    $smarty->assign('navigation3',$navigation3);
    $smarty->assign('navigation4',$navigation4);
    $smarty->assign('navigation5',$navigation5);
    $smarty->display("templates/index.html");
?>
```

上述代码试用声明变量将通过使用 assign()方法将数据绑定到变量内，最后通过使用 Smarty 中的 display()方法转向显示模板 index.html 页面。

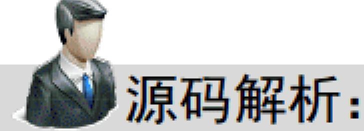
11.3.3 运行结果

保存页面信息启动服务运行 index.php 页面，运行结果如图 11-3 所示。



图 11-3　网站简介

11.3.4　实例分析



源码解析：

在实例中，主要讲解了 Smarty 模板的配置和简单的使用，在代码中 template_dir 是用来放置 Smarty 编译的所有模板的位置，而 caching 则是用来设置是否关闭缓存，true 是开启缓存，使用 display()方法将页面转向模板页面。

11.4　新闻信息列表

对于很多读者来说，对 Smarty 等模板引擎通常抱怨说引擎的特性集集成了某种程度的逻辑。毕竟模板引擎的宗旨就是完全分离表示层和业务逻辑层，虽然这是我们的理想，但不一定是最可行的解决方案。比如，如果没有某种迭代逻辑，如何以某种格式输出数据库结果？要得到完美的解决方案是不可能的。面对这个两难的问题，Smarty 开发人员在引擎中集成了一些简单而且非常有效的引用程序逻辑。本节将会学习 Smarty 的表现特性，包括变量修饰符、流程控制和语句。

视频教学：光盘\videos\11\新闻信息列表.avi　　　　长度：14 分钟

11.4.1 基础知识——Smarty 模板注释

在本章讲解过程中必要时都会使用注释。因此，在这里我们首先讲解一下 Smarty 的注释语法，在这里只简单介绍。注释包围在定界“ {*” 和 “*} ” 之间，可以包括单行或者多行。以下代码是正规的 Smarty 注释语法，代码如下所示。

```
{*$navigation3="技术文档";*}
```

11.4.2 基础知识——变量修饰符

在之前的章节中我们已经看到，PHP 提供了很多函数，这些函数能够以你能想象到的任何方式处理文本数据。但是你可能想在表现层中使用这些特性。因此，Smarty 开发人员在库中已经为我们继承了许多表现特有的功能。

Smarty 是一个不太传统的变量修饰符语句，当然，定界符用来指示所请求的变量输出，而在输出前，需要修饰的变量值后面可以跟一个竖线，在后面是修饰符命令，基本语法如下所示：

```
{$var|modifier}
```

上述基本语法将在本节讲解修饰符语法中反复用到。

1. capitalize 函数

capitalize 函数把变量中所有单词的首字母变为大些字母，该函数语法如下所示：

```
$smarty = new Smarty;
$smarty->assign("title","窗内网");
$smarty->display("index.html");
```

在 index.html 显示数据页面中包含以下代码：

```
{$title|capitalize}
```

2. count_words 函数

count_words 函数用来统计变量中的单词总数，详细代码如下所示：

```
$smarty = new Smarty;
$smarty->assign("title","窗内网");
$smarty->assign("body", "免费视频教程基地");
$smarty->display("index.html");
```

在 index.html 页面内编写以下代码语法：

```
<strong>{$title}</strong>(共有{$body|count_words}字符)分别是
<p>{$body}</p>
```

3. date_format 函数

date_format 函数是 PHP 中 strftime()函数的包装器，它可以将 strftime()函数获得的任何日

期/时间格式字符串转换为某种特殊格式。因为格式化标签在之前章节中已经讲解到了，在这里就不再讲述，以下代码是使用 date_format 函数格式化时间：

```
$smarty = new Smarty;
$smarty->assign("filed","20110101");
$smarty->display("index.html");
```

在页面模板中包含以下代码：

```
{$filed,"%B%e%Y"}
```

4．default 函数

当应用层没有返回结果时，通过使用 default 函数可以指示特定变量的默认值提供了一种简单的方式，代码如下所示：

```
$smarty = new Smarty;
$smarty->assign("title","窗内网");
$smarty->display("index.html");
```

在模板页面书写以下代码：

```
{$body|default:"暂时没有数据显示。"}
```

5．strip_tags 函数

strip_tags 函数主要作用删除变量字符串中的标识符，比如在变量字符串中存在着大量的标签符号，而保存数据库时不希望这些标签符号也保存到数据库内，那么使用该函数可以过滤掉这些标签，代码实例如下所示：

```
$smarty = new Smarty;
$smarty->assign("body","<p>窗内网</p>");
$smarty->display("index.html");
```

显示层页面嵌入以下代码：

```
{$body|strip_tags}
```

就这样，当页面输出信息时将会把变量中的字符串 p 标记过滤掉。

6．truncate 函数

该函数将变量字符串截取为指定数量的字符串。虽然默认为 80 个字符，但可以通过提供一个输入参数来改变截取字符串的长度。可以指定一个字符串，追加到截取字符串的后面，如省略符号(……)。除此之外，可以指定到达指定的字符串数的长度后立即截取，或者还需要考虑单词的边界。



参数为 true，则按照确切的长度截取，false 则截取达到长度后最近的单词边界。

下面代码是从一个列表中读取数据，限制截取每条数据显示字符串的长度，详细代码如下所示：

```
$summaries = array(
    "一、用户在本站发布的内容并不反映任何窗内网之意见",
    "二、窗内网尊重并保护所有使用本站用户的个人隐私权，用户注册的用户名、电子邮件地址等个人资料",
    "三、窗内网尊重并保护所有使用本站用户的个人隐私权，用户注册的用户名、电子邮件地址等个人资料",
);
$smarty = new Smarty;
$smarty->assign( "summaries","$summaries");
$smarty->display("index.html");
```

在上述代码中，定义了一个数组变量，将信息内容通过数组的形式保存在变量中，然后在转向表示层页面以下是表示层中嵌套代码：

```
{foreach fom=$summaries item=$summary}
    {$summary|truncate:15:"…."|false}<br/>
{/foreach}
```

在代码中，通过使用循环标签将数组中的数据循环显示出来，然后通过使用 truncate 函数截取每条数据只显示 15 个字符，在后面使用省略号追加。

11.4.3 基础知识——流程控制结构

对于程式执行的流程顺序以及应因不同状况而选取不同的流程，即为流程控制(flow control)。流程控制可说是计算机程式的灵魂，它包含条件判断、回圈控制及无条件跳跃三大类。

1. if-elseif-else 函数

Smarty 的 if 语句与 PHP 语言中的 if 语句相同，与 PHP 一样，可以使用一些条件限定符，如表 11-1 所示。

表 11-1 PHP 条件限定符

eq	gt	gte	ge
lt	lte	le	ne
neq	is even	is not even	is odd
is not odd	div by	even by	not
mod	odd by	==	!=
<	>	<=	>=

下面是一个简单的实例代码：

```
{*assume $dayofweek=6.*}
{if $month<5}
    <p>今天是周末</p>
{/if}
```

可能有的时候通过代码不够直观地观察出流程语句的语法执行顺序，在图 11-4 中直观地

看到 if 流程语句的执行过程。

再下面看另一个实例。假如希望根据月份插入某条数据信息，需要使用条件限定符、if、elseif 以及 else 语句来完成这项任务，代码如下所示：

```
{if $month < 4}
    夏天到来了
{elseif $month ge 4&& $month<=9}
    这几个月会很热！
{else}
    热的感觉！
{/if}
```

同样 if、elseif 以及 else 语句同样可以使用流程图来表示如图 11-5 所示。

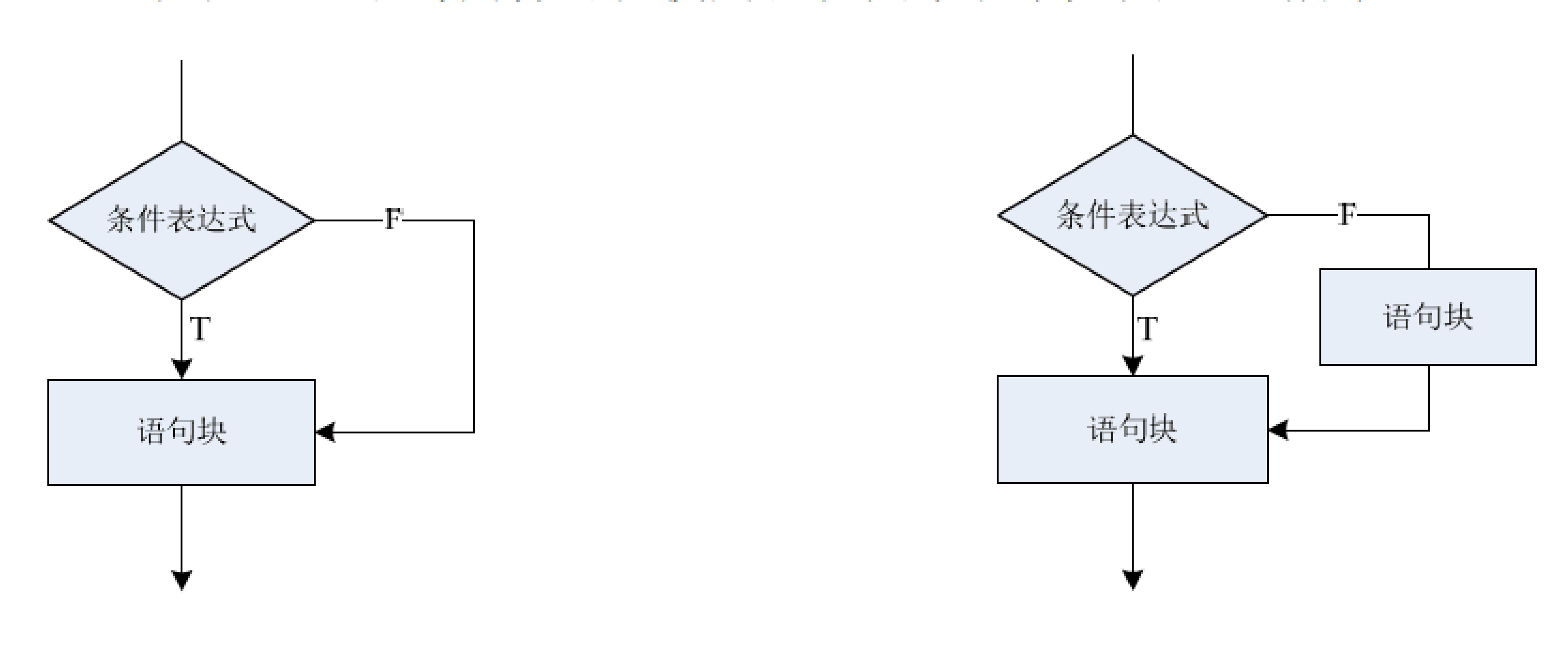



图 11-4　if 流程图

图 11-5　if、elseif 以及 else 语句块流程图



把条件语句包围在小括号中是可选的，但在标准的 PHP 代码中这确实是必须的。

2．foreach 函数

foreach 标记的作用与 PHP 语句中的命令相同。但如下所示，其语法大不相同。它有 4 个参数，分别如下所示，其中有两个参数是必须的。

- from：这个必要参数指定目标数组的名。
- item：这个参数也是必要的测试确定当前元素的名。
- key：这个可选参数确定当前键的名。
- name：这个可选参数确定节的名。这个名称是任意的，应当设置为一个描述性的名称。

在之前的实例中也用到了 foreach 语句，在这里通过使用循环处理一周中的每一天。

```
require(Smarty.class.php);
$smarty = new Smarty;
$daysofweek = array("星期日","星期一","星期二","星期三","星期四","星期五","星期六
");
$smarty->assign("daysofweek",$daysofweek);
$smarty->display("index.html");
```

在页面显示列表中使用 foreach 标记将数组中的列表循环遍历到页面内，代码如下所示：

```
<body>
```

```
    <select name="select" id="select">
            {foreach from=$daysofweek item=day}
        <option value="{$day}">{$day}</option>
            {/foreach}
    </select>
</body>
```

同样使用 foreach 标记可以实现使用 key 属性迭代处理一个关联数据，比如省市区的名称的关联菜单，详细代码如下所示：

```
require(Smarty.class.php);
$smarty = new Smarty;
$states = array("河南"=>"郑州","河北"=>"唐山","湖南"=>"长沙","云南"=>"昆明");
$smarty->assign("daysofweek",$daysofweek);
$smarty->display("index.html");
```

如上述代码数组中存取的数据格式，使用=>符号将有管理的数据连接起来，然后使用 foreach 标记循环显示在页面上，代码如下所示：

```
{foreach from=$states item=day item=item}
        {$key}: {$item}<br/>
{/foreach}
```

通过上述实例，运行后将会在页面显示如下结果：

```
河南：郑州
河北：唐山
湖南：长沙
云南：昆明
```

虽然 foreach 语句非常有用，但绝对需要花许多时间学习另一个与之类似的 section 语句。该语句的功能非常强大，在后面我们会讲解到。

3．foreachelse 函数

foreachelse 标记与 foreach 一起使用，与作用于字符串的 default 标记作用类似，数组为空时，foreachelse 标记可以生成某个候选结果。以下代码是一个使用 foreachelse 的模板实例：

```
{foreach key=key item=item from=$titles}
    {$key}: {$item}<br/>
{foreachelse}
    <p>抱歉，没有找到您想要的信息！</p>
{/foreach}
```



foreachelse 不使用结束符号。它嵌入 foreach 中，与 elseif 嵌入到 if 语句中的性质类似。

4．section 函数

section 函数的操作就像一个改进的 for 或 foreach 语句，它会迭代处理并输出数据数组，但其语法差异很大。这里“改进”一词是指它与 for 或 foreach 语句结构提供了相同的循环特性，

另外还提供了很多附加选项，可以更多地控制循环的执行。这些选项都要通过函数参数来支持。以下是该语句的参数信息。

- name：确定节的名称。名称可以任意，应当设置为能够描述节的目的。
- loop：设置循环迭代的次数，应设置为与数组变量同名。
- start：确定迭代开始的所引位置。例如，如果数组包含 5 个值，而 start 设置为 3，则迭代将从数组的所引 3 开始。如果给出的是负值，则开始位置由从数组末尾减去该数字来确定。
- step：确定在数组中移动的步长值。该属性的默认值为 1。例如，设置 step 为 3 将导致迭代在数组所引 0、3、6、9 等等发生。设置 step 为负值则导致迭代数组从数组末尾向前进行。
- max：确定迭代的最大次数。
- show：确定是否显示此节。可以使用这个参数进行调试，然后在部署时再把这个参数设置为 false。

下面我们通过简单的实例来介绍这些参数的作用，代码如下：

```
require("Smarty.class.php");
$smarty = new Smarty;
$daysofweek = array("星期一","星期二","星期三");
$smarty->assign("daysofweek",$daysofweek);
$smarty->display("index.html");
```

接下来是在显示模板内，使用 section 标记循环遍历数据，代码如下所示：

```
{secton name=day loop=$daysofweek}
    {$ daysofweek [day]}<br/>
{/section}
```

在这个有些奇怪的语法中，注意，节名必须像数组中的索引值一样引用。而且还要注意的是，$ daysofweek 变量名有双重职责，它既是循环指示器，也是实际的变量引用。

看一下使用管理数组的实例，代码如下所示：

```
require("Smarty.class.php");
$smarty = new Smarty;
$states[] = array("ha"=>"郑州","hb"=>"唐山","hun"=>"长沙","yn"=>"昆明");
$states[] = array("ha"=>"三门峡","hb"=>"保定","hun"=>"岳阳","yn"=>"丽江");
$smarty->assign("states",$ states);
$smarty->display("index.html");
```

下面是通过使用 section 的另外一种分类形式将数据循环显示，代码如下所示：

```
{section name=sm loop=$states}
    <p
    河南：{$states[sm].hn}<br/>
    河北：{$states[sm].hb}<br/>
    湖南：{$states[sm].hun}<br/>
    云南：{$states[sm].yn}</p>
{/section}
```

通过上述实例运行后返回的页面结果如下所示：

```
<p>
    河南：郑州
    河北：唐山
    湖南：长沙
    云南：昆明
</p>
<p>
    河南：三门峡
    河北：保定
    湖南：岳阳
    云南：丽江
</p>
```

5．sectionelse 函数

sectionelse 函数与 section 一起使用，该函数执行原理和上面讲到的 foreachelse 函数非常相似，数组为空时它生成某个候选数据，下面是使用 sectionelse 的模板实例代码：

```
{section key=key item=item from=$titles}
    {$key}: {$item}<br/>
{sectionelse}
    <p>抱歉，没有找到您想要的信息！</p>
{/section}
```

注意，sectionelse 不使用结束括号，相反，它同样是嵌入使用的语法，它与 section 嵌套使用，像之前讲解到的 foreachelse 嵌入在 foreach 语句中一样。

11.4.4　基础知识——Smarty 语句

Smarty 为大家提供了几个用于完成特殊任务的语句，比如 include 用于导入模板使用，insert 则是用于导入缓存数据，还有 literal 和 php 等语句，这节将介绍这些语句的使用方法和它们的作用。

1．include 语句

include 语句与 PHP 包中的同名语句相同，只是它只用于将其他模板导入到当前模板。例如，假设希望在 Smarty 模板中导入两个文件 header.tpl 和 footer.html，可以如下完成：

```
{include file="/lib/tpl/header.tpl"}
{*在这里执行其他 Smarty 语句*}
{include file="/lib/html/footer.html"}
```

这个语句还提供了另外两个特性。首先，可以传入可选属性 assign，将所导入文件的内容赋给 assign 指定的变量。代码如下所示：

```
{include file="/lib/header.tpl;" assign="header"}
```

使用上述代码导入文件，这样一来，不会输出 header.tpl 里面的内容，而是输出指定的

$header 变量中赋的值。

第二个特性允许向所有导入文件传递各种属性。例如，假设希望将属性 title=“窗内网登录系统”传给 login.html 文件，那么代码如下所示：

```
{include file="/html/login.html" title="窗内网登录系统"}
```

在这里大家需要知道，以上传递的任何属性只能在所导入文件的范围内使用，不能适用于模板的其他位置。



fetch 语句与 include 完成相同的任务，都是将文件嵌入模板中，但它们有两点不同，首先，除了获取本地文件外，fetch 还可以使用 HTTP 和 FTP 协议获取文件；其次，fetch 没有相关选项在获取文件时指定属性的。

2. insert 语句

insert 标记与 include 标记的功能相似，唯一不同的就是他要导入不会被缓存的数据。比如，可以使用一个函数插入经常更新的数据，比如实时新闻、天气预报或者其他在这段时间内需要更新数据的内容。它也接受以下几个参数。

- name：用来确定 insert 函数名。
- assign：用于将输出赋给变量，而不是直接发送到输出。
- script：用来指定在导入文件前直接执行的一个 PHP 脚本。当输出文件的内容依赖于脚本所完成的某个特定动作时，可以使用此阐述。
- var：该参数用于传入所有插入模板使用的其他参数。

name 参数很特殊，用于指定插入语句所插入内容的特定命名空间。当遇到 insert 标记时，Smarty 将调用一个名为 insert_name()的用户定义 PHP 函数，通过函数的 var 参数传入 insert 标记中的所有变量。无论这个参数返回的输出是什么，都将会放在 insert 标记所在的位置。

下面示例中，使用动态加载网站公告，让用户完成实时观看最新的公告内容，代码如下所示：

```
function insert_banner()
{
    $query="select ms from message";
    $result=mysql_query($query);
    return mysql_result($result,0,ms);
}
```

然后将查询出来的数据显示在页面中，代码如下所示：

```
<div>
公告：{insert name="banner"}
</div>
```

遇到以上语句时，Smarty 将引用所有可用的名为 insert_banner()的用户定义 PHP 函数，同样还可以使用该函数来传递参数内容。

3. literal 语句

literal 语句是用来使 Smarty 所有嵌入的任何数据都原样输出，不需要进行转换。该标记主

要用于模板嵌入 Javascript 和 CSS 中，从而不需要担心与 Smarty 的定界符冲突。使用语法如下所示：

```
<html>
    <head>
        <title>用户登录</title>
        {literal}
            <style type="text/css">
        div{margin:1px;}
            </style>
        {/literal}
    </head>
…
</html>
```

在使用的过程中，如果没有将 CSS 或者 Javascript 信息包含在 literal 括号中，会导致 Smarty 生成解析错误，因为它会去尝试解析 CSS 标记中的大括号，认为是 PHP 中的定界符。

4．PHP 语句

该语句可以使用 PHP 函数在模板中嵌入 PHP 代码，{php}{/php}标记中的任何代码都由 PHP 引擎处理。以下代码是使用该 PHP 的嵌入基本语法。

```
<html>
    <head>
        <title>用户登录</title>
    </head>
    <body>现在时间为：{php}echo date("Fj,Y");{/php}</body>
</html>
```

通过以上实例将会输出以下信息：

```
December 1,2011
```



在 PHP 的函数中 include_php()在模板导入函数 PHP 代码的单独脚本，从而更清晰地分离代码。该函数还可以使用其他选项。

11.4.5　实例描述

随着互联网的进一步发展，网络媒体在人们心中的地位进一步提高，新闻发布系统作为网络媒体的核心系统，其重要性是越来越突显：一方面，它提供一个新闻管理和发布的功能；另一方面，现在的新闻发布要求与普通的用户实现交互。该实例通过使用新闻系统中的新闻列表为实例进行讲解，使用 Smarty 模板技术来完成，新闻信息列表是将站内的所有信息以列表的形式呈现在用户面前，使得用户清晰而又方便地直观阅读站内的新闻信息。

11.4.6　实例应用

【例 11-3】 新闻信息列表

该实例主要通过使用 Smarty 模板来完成，那么首先需配置 Smarty 模板环境，其次是编写模板创建存储 Smarty 模板和配置文件的 4 个目录，最后创建 smarty.php 文件定义 Smarty 的$template_Dir、$compile_dir、$config_dir 和$cache_dir 修改默认行为，代码如下所示。

```
<?php
include("smarty/libs/Smarty.class.php");
    $smarty = new Smarty;
    $smarty->template_dir = 'templates';
    $smarty->config_dir = 'config';
    $smarty->cache_dir = 'smarty_cache';
    $smarty->compile_dir = 'smarty_templates_c';
    //上面四行为使用 Smarty 前的必要参数配置
    $smarty->caching=false;         //关闭缓存
    $smarty->left_delimiter="{";    //左边界
    $smarty->right_delimiter="}";   //右边界
?>
```

在上述代码中，使用 include()方法将 Smarty 模板核心文件引入页面内，设置 Smarty 的$template_Dir、$compile_dir、$config_dir 和$cache_dir 修改默认行为，而 caching 属性用来设置是否缓存的功能，left_delimiter 和 right_delimiter 则用来设置左右定界符的功能。

接下来创建 conn.php 页面用来连接数据库，该文件主要功能则是直接与数据库进行连接，打开关闭数据库，代码如下所示：

```
<?php
        $conn=mysql_connect("localhost","root","pwd") or die("数据库服
务器连接错误".mysql_error());
        mysql_select_db("news_db",$conn) or die("数据库访问错误".mysql_error());
        mysql_query("set character set gb2312");
        mysql_query("set names gb2312");
?>
```

在上述代码中，mysql_connect()函数的主要作用就是打开或重复使用一个到 MySQL 服务器的非持久性连接，如果要创建持久连接还需要使用 mysql_pconnect()函数。mysql_connect()函数中有 3 个参数，第一个参数 localhost，代表连接数据库服务器名称，第二个参数 root 用来设置连接 MySQL 数据库的账号，而第三个参数 pwd 用来设置连接数据库的密码。以上是打开数据库连接，接下来使用 mysql_select_db()函数来获取需要连接数据库的名称。mysql_query()函数的作用是用来转换从数据库内获取数据库的编码格式，以便读取数据乱码行为。

到这里数据已经连接成功了，接下来进行读取数据，将数据从数据库内读取出来进行业务处理，代码如下所示。

```
<?php
    include("conn/conn.php");
    include("smarty.php");
    $rows=array();
    $sql=mysql_query("select * from newscontent",$conn);
    while($row=mysql_fetch_assoc($sql))
    {
```

```
        array_push($rows,$row);
    }
```

在上述代码中，首先使用 include()方法嵌入了两个页面 conn.php 和 smarty.php，conn 文件则是用来连接数据库用的文件，在前面编写好的连接数据库代码则可以直接应用到该文件中。smarty 文件是配置 Smarty 模板属性的文件，在该文件中，定义了一些相关的 Smarty 模板函数值等。接下来使用 mysql_query()方法来获取数据数据信息，该方法有两个参数，其中一个代表执行的 SQL 语句，另外一个是打开数据库连接。最后将数据在 while 循环中使用 mysql_fetch_assoc()方法读取数据保存至数组中。下面是将数组转向模板页面，代码如下所示：

```
    $infos=$rows;
    $htmltitle="新闻信息列表";
    $navigation = array("文化新闻","房产新闻","财经新闻","体育新闻","经济新闻","社
        会新闻","国际新闻报","国内娱乐新闻");
    $smarty->assign("htmltitle",$htmltitle);
    $smarty->assign("navigation",$navigation);
    $smarty->assign("infos",$infos);
    $smarty->display("newlist.html");
?>
```

在上述代码中，声明页面 htmltitle 变量将标题保存至该变量中，然后声明数组将导航信息保存在数组中，使用 assign()方法，将数据传入到模板页面中，最后使用 display()方法，转向模板页面。下面是模板页面 newslist.html 中的代码：

```
{section name=key loop=$navigation}
        <A style="COLOR: #fff"
href="http://localhost:8080/News/userinfo.do?o=newsmsbytypeid&amp;typeid=25
">{$navigation[key]}</A>  |
        {/section}
```

上述代码中使用 section 标记将导航数组中的导航信息通过循环的方式显示在页面中，section 函数的操作就像一个改进的 for 或 foreach 语句，它会迭代处理并输出数据数组，但其语法差异很大。下面代码是用来显示新闻列表信息：

```
{foreach from=$infos item=info}
    <TABLE border=0 width="100%">
        <TBODY>
        <TR>
        <TD style="FONT-SIZE: 14px" width="100%">    
<A  title="" href="http://localhost:8080/News/userinfo.do?o=newsmsbyid&amp;
id=119">
<STRONG>{$info['newstitle']|truncate:20:"":true}</STRONG></A></TD></TR>
        <TR>
        <TD>  日期： {$info['time']} 点击：<FONT
color=#ff0000> {$info['click']} </FONT>  </TD></TR>
        <TR>
        <TD><div style="width:650px;line-height:25px;
text-overflow:ellipsis; white-space:nowrap; overflow:hidden;">
        {$info['content']|truncate:110:"...":true}
```

```
</div></TD></TR></TBODY></TABLE>
       <HR align=center>
   {foreachelse}
       <p>抱歉，没有找到相应的信息！</p>
    {/foreach}
```

在代码中，通过使用 foreach 标记来完成新闻列表的显示，该标记与 PHP 中的 foreach 非常相似，有 4 个参数，在这里只使用了 from 属性，该属性用来指定目标数组的名称，在代码中我们设置为 infos，item 属性确定当前元素的名称。在代码中还使用了 truncate 函数截取字符串，当该字符串的内容超过指定长度时截取，使用指定字符串替换。

11.4.7　运行结果

到这里，新闻信息列表系统已经大致完成了，接下来运行业务处理页面，运行结果如图 11-6 所示。



图 11-6　新闻消息列表

11.4.8　实例分析



源码解析：

通过上述实例的讲解，大家一定对 Smarty 模板的使用有了更加深刻的印象。在实例中主要使用了 Smarty 模板的流程控制语句 foreach 和 section 来控制从数据库内读取的显示，当内容显示在模板页面时，通过使用模板中的一些函数对数据进行处理。

11.5 个人博客

Blog是一个简单交流类型的网站，通常有简短且经常更新的帖子，由个人管理，可不定期发表新的文章。Blog的内容和目的有着很大的不同，例如评论、个人的日记、照片、诗歌、散文，甚至科幻小说的发表或心情等都有。许多Blog是个人心中所想之事的发表，其他Blog则是一群人基于某个特定主题或共同利益领域的集体创作。因此本节采用Smarty模板来实现个人博客功能信息。

11.5.1 实例描述

个人博客的功能想必大家一定比我更加熟悉了，因此在这里不多啰嗦，该实例仅仅实现一个文章展示功能，将数据库中的文章信息读取到页面中，通过使用foreach循环标记将内容显示在前台页面中，在页面的右侧显示文章的类型，以及该类型下的最新文章信息。要求使用Smarty模板动静分离形式实现该实例。

11.5.2 实例应用

【例 11-4】 个人博客

在制作该实例之前，同样需要搭建Smarty环境在我们的工程目录下，其次则是编写模板创建存储Smarty模板和配置文件的4个目录，编写smarty.php模板文件，代码如下所示：

```
<?php
include("smarty/libs/Smarty.class.php");
    $smarty = new Smarty;
    $smarty->template_dir = 'templates';
    $smarty->config_dir = 'config';
    $smarty->cache_dir = 'smarty_cache';
    $smarty->compile_dir = 'smarty_templates_c';
    //上面四行为使用Smarty前的必要参数配置
    $smarty->caching=false;          //关闭缓存
    $smarty->left_delimiter="{";     //左边界
    $smarty->right_delimiter="}";    //右边界
?>
```

代码中，首先通过使用include()方法将Smarty.class.php核心文件引入该模板页面中，创建新的Smarty对象。使用该对象下的4个函数设置相关的配置，然后是定义模板页面的左右定界符。

Smarty模板创建好了，接下来是连接数据库文件conn.php，通过Smarty的内置函数来打开数据库连接，代码如下所示：

```
<?php
    $conn=mysql_connect("localhost","root","root") or die("数据库服务器连接错误
".mysql_error());
    mysql_select_db("php_blog",$conn) or die("数据库访问错误".mysql_error());
    mysql_query("set character set gb2312");
    mysql_query("set names gb2312");
?>
```

上述代码中，使用 mysql_connect()方法连接服务器，然后通过使用 mysql_select_db()方法访问数据库。如果连接成功，接下来是对数据库进行操作，将数据库内的数据读取到 PHP 页面中。下面创建 index.php 页面处理业务逻辑读取数据内容，代码如下所示：

```
<?php
    include("conn/conn.php");
    include("smarty.php");
    $top5blogrows=array();  //前 5 条文章数组
    $blogrows=array();      //所有文章数组
    $typerows=array();      //所有文章类型数组
    $top5sql=mysql_query("select * from content limit 5",$conn);
    //前 5 条文章 SQL 语句
    $sql=mysql_query("select * from content",$conn);        //所有文章 SQL 语句
    $typesql=mysql_query("select * from blogtype",$conn);  //文章类型 SQL
    //循环绑定前 5 条文章
    while($top5blogrow=mysql_fetch_assoc($top5sql))
        {
            array_push($top5blogrows,$top5blogrow);
        }
    //循环绑定所有文章
    while($blogrow=mysql_fetch_assoc($sql))
        {
            array_push($blogrows,$blogrow);
        }
    //循环绑定文章类型
    while($typerow=mysql_fetch_assoc($typesql))
        {
            array_push($typerows,$typerow);
        }
    $htmltitle="窗内网博客";
    $smarty->assign("htmltitle",$htmltitle);   //博客标题
    $smarty->assign("infos",$top5blogrows);    //前 5 文章数据
    $smarty->assign("cinfos",$blogrows);       //所有文章数据
    $smarty->assign("tinfos",$typerows);       //文章类型
    $smarty->display("index.html");
?>
```

数据库连接成功后，就要对数据库内数据进行读取。首先在代码中使用 include()方法将 smarty.php 文件和 conn.php 文件引入当前目录内，提供对数据库的操作，和 Smarty 模板的使用。在代码中创建了 3 条 SQL 语句，分别用来对不同数据进行读取，并且保存在数组集合中，然后转向静态模板页面 index.html 中进行循环展示，静态页面代码如下所示。

```
<div class="main">
        {foreach from=$infos item=info}
        <div class="item">
            <div class="date">
                <div>OCT</div>
                <span>01</span>
            </div>
            <div class="content">
                <h1>{$info['blogtitle']|truncate:45:"...":true}</h1>
                <div class="body">
                    <p class="contentfont">     {$info
                    ['blogcontent']}
                </p>
            </div>
            </div>
        </div>
        {foreachelse}
        <p>抱歉，没有找到相应的信息！</p>
         {/foreach}
    </div>
```

在代码中通过使用 foreach 语句将数组 infos 中的数据展现在页面中，使用{$info['blogcontent']}将数据绑定在页面中。在标题绑定时使用 truncate 将数据截取较长字符串到指定长度。接下来是页面右侧文章类型显示代码。

```
<div class="navigation">
    {foreach from=$tinfos item=tinfo}
        <h1>{$tinfo['typename']}</h1>
        <ul>
        {foreach from=$cinfos item=cinfo}
            {if $tinfo['id']==$cinfo['typeid']}
            <li><a href="index.php" class="contentfont">
{$cinfo['blogtitle']|truncate:25:"...":true}</a></li>
            {/if}
        {/foreach}
        </ul>
        {foreachelse}
        <p>抱歉，没有找到相应的信息！</p>
    {/foreach}
    </div>
```

同样，在这里使用 foreach 语句将文章类型循环展示在页面中，然而唯一不同的是在这里我们还使用了 if 语句，当文章表中的文章类型 ID 和类型表中 ID 相等时，将该类型下的最新文章暂时显示在该类型下面的子级菜单中。

11.5.3　运行结果

打开浏览器，在地址栏中输入 http://localhost/ch5/index.php，运行效果如图 11-7 所示。



图 11-7　个人博客首页

11.5.4　实例分析



源码解析：

在上述实例中，将前面所学的 Smarty 模板技术全部融入该实例中进行讲解，通过使用 foreach 语句循环数组集合中的数据信息到页面中，在通过 if 语句进行数据业务处理。当标题内容过多时，可使用 truncate 将字符串进行截取，从而完成个人博客的功能。

11.6　常见问题解答

11.6.1　使用 Smarty 模板将数据显示在模板时出现乱码



使用 Smarty 模板出现乱码怎么办？

网络课堂：http://bbs.itzcn.com/thread-13744-1-1.html。

今天使用 PHP 中 Smarty 完成一个用户列表系统，当输出通过该模板显示在页面时，总是出现乱码现象，而且静态页面中的编码格式为 UTF-8，这是什么原因？

【解决办法】：造成乱码的原因有很多种，比如说，您的 Smarty 模板的文件格式不是 UTF-8 编码格式，或者在获取数据库信息时，没有进行转码。在读取数据库数据时使用以下代码进行转码：

```
mysql_query("set character set gb2312");
mysql_query("set names gb2312");
```

11.6.2　无法获取 Smarty 模板数据



在静态页面中，无法获取 Smarty 模板提交的数据内容

网络课堂：http://bbs.itzcn.com/thread-13747-1-1.html

第一次使用 Smarty 模板时，很容易就会犯的一个错误是无法获取 Smarty 模板中的数据内容，这让初学 Smarty 模板的读者们非常头痛，今天我就遇到了这样的问题，该如何解决？

【解决办法】：使用 Smarty 模板无法获取数据主要有以下几种：首先我们需要看 Smarty 环境是否配置成功，如果配置没有什么问题，则看转向页面是否使用 assign()方法将数据内容保存起来，最后则需要注意使用 display()方法转向页面。

11.7　习　　题

一、填空题

(1) include_path()方法用法为设置一个名为__________的预定义常量，指向 Smarty 类库路径，然后类库名以常量开头。

(2) Smarty 模板中，可以分别使用 Smarty 的$template_Dir、$compile_dir、___________和$cache_dir 修改默认行为。

(3) __________函数把变量中所有单词的首字母变为大写字母。

(4) __________函数是 PHP 中 strftime()函数的包装器，它可以将 strftime()函数获得的任何日期/时间格式字符串转换为某种特殊格式。

(5) 参数为 TRUE，则按照确切的先知截取，__________则截取到达到先知后最近的单词边界。

(6) __________标记的作用与 PHP 语句中的命令相同。

(7) __________标记与 foreach 一起使用，与作用域字符串的 default 标记作用类似，数组为空时，foreachelse 标记可以生成某个候选结果。

(8) ___________标记与 include 标记的功能相同，唯一不同的就是他要导入不会被缓存的数据。

二、选择题

(1) 以下选项中哪个不属于 Smarty 模板配置文件夹__________。

A. templates　　B. configs　　C. templates_c　　D. src

(2) 哪个选项属于正确使用 templates_c()函数的__________。

A.

```
{foreach fom=$summaries item=$summary}
  {$summary|truncate:15:"…."|false}<br/>
{/foreach}
```

B.

```
{foreach fom=$summaries item=$summary}
  {$summary|truncate:15}<br/>
{/foreach}
```

C.

```
{foreach fom=$summaries item=$summary}
  {$summary:"…."|false}<br/>
{/foreach}
```

D.

```
{foreach fom=$summaries item=$summary}
  {$summary|truncate|15|"…."|false}<br/>
{/foreach}
```

(3) foreach 标记的作用与 PHP 语句中的命令相同，那么以下选项中，哪个选项不属于该标记的属性__________。

A. key　　B. item　　C. id　　D. name

(4) insert 标记与 include 标记的功能相同，唯一不同的就是他要导入不会被缓存的数据，以下选项中，属于该标记的属性为__________。

A. url　　B. script　　C. id　　D. item

三、上机练习

上机练习： 图书推荐。

使用 PHP 中的 Smarty 模板来实现图书推荐信息显示功能，要求从数据库内读取推荐图书名称、图片和简介等信息，使用 foreach 标记将数据显示在页面中，运行结果如图 11-8 所示。



图 11-8　图书推荐



第 12 章　在 PHP 中处理 XML

内容摘要：

XML(eXtensible Markup Language)是一种可扩展标记语言，用于存储数据，并且能够使数据通过网络无障碍地进行传输。XML 还允许用户创建和使用自己的标记来描述要表达的内容，且它注重的是本身的格式和数据内容。

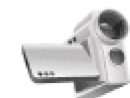
本章将对 XML 的基础知识进行简单介绍，重点讲述如何使用 PHP 处理 XML，包括 SAX、DOM 以及 SimpleXML 这三种方式的使用方法及区别。

学习目标：

- 了解 XML 文档形式。
- 掌握 XML 文档结构。
- 了解 PHP 解析 XML 的各个方法的优缺点。
- 掌握 SAX 解析器解析 XML 文档。
- 掌握使用 DOM 库操作 XML 文件。
- 掌握 SimpleXML 处理 XML 文件。
- 了解客户端处理 XML 文件。

12.1 创建用于存储图书信息数据的 XML 文件

XML 是 Internet 环境中跨平台且依赖于内容的技术，是当前处理结构文档信息的最佳工具。这使 XML 更容易与 Windows、Mac OS、Linux 以及其他平台产生信息结合。本节将介绍 XML 的优点、XML 文档结构及如何创建 XML 文档。

 **视频教学：**光盘\videos\12\XML 基本知识.avi　　　　长度：7 分钟

12.1.1 基础知识——XML 概述

由于 XML 是以文本形式来描述的一种文件格式，所以 XML 可以跨越不同平台进行正常的数据交互。但是，文本形式也会因为文字编码的不同出现不能阅读的问题，在这一点上 XML 有着非常完美的解决方案。

HTML 由于难以扩展、交互性差、语义性差以及超链接单向等缺点，使其在电子数据交换、数据库与搜索引擎等领域的应用存在着障碍，而 XML 的出现也是为了解决这些问题。XML 是一种元标记语言，并且允许用户自定义，但是这些标记必须根据某些通用的原理来创建。任何系统都可以通过 XML 的解析器来读取 XML 数据，因此 XML 数据可以随处执行，而不用担心出现系统不支持的问题。总体来说，XML 与 HTML 相比具有如下的优点。

- 开放性：允许在任何平台上读取和处理数据，更重要的是 XML 允许通过 HTTP 和其他传输协议交换 XML 数据。
- 简单性：XML 文档为纯文本文件，这样 XML 文档就可以自由地在两个不同的系统之间交换数据，因为基本上任何系统都提供了对文本格式的支持。
- 自我描述性：XML 可以明确地表示数据模型中各个部分的意义，这就是自我描述性，只要查看 XML 文档就能确定该数据的意义。
- 互操作性：XML 文档是纯文本的，很容易在系统间传递。而且 XML 更容易在应用程序间迁移，不同的环境和操作系统可以互相通信。
- 可扩展性：使用 XML 可以根据需要定制语言，每种语言都用于特定的环境。

XML 是一种基于文本格式的文本文件，每一个 XML 元素包括一个开始标签和一个结束标签，以及两个标签之间的信息。XML 允许有无限的标签集，XML 各个标签描述的是文档的结构和意义，不会描述页面元素的格式。

另外，任何一个格式定义良好的 XML 文档必须包含一个根，跟在处理指令或是 DTD 定义的后面，如果没有处理指令或是 DTD 文档，则跟在 XML 文件声明的后面。下面来看一个最简单的 XML 文件，代码如下所示：

```
<?xml version="1.0" encoding="GB2312"?>
<!--下面是 XML 的元素-->
<book>
    <bookname>Web 成功开发之路之 PHP</bookname>
    <time>2011-01-01</time>
```

```
    <author>马向林</author>
</book>
```

在上面的代码中，第一行表示声明 XML，第二行表示 XML 文档中的一行注释。接下来声明了 XML 的根元素，并在根元素中定义了 3 个子元素。



XML 文档具有严格的格式要求，严格区分大小写，每个元素必须被关闭，并且关闭的顺序必须和定义顺序保持一致。

12.1.2　基础知识——XML 文档结构

创建一个 XML 文档时，首先需要添加完整的声明格式，以告知浏览器或者其他处理程序，该文档为 XML 文档。添加完声明格式之后，就可以为 XML 文档添加处理指令、注释和实体等，图 12-1 所示为一个详细的 XML 文档结构图。

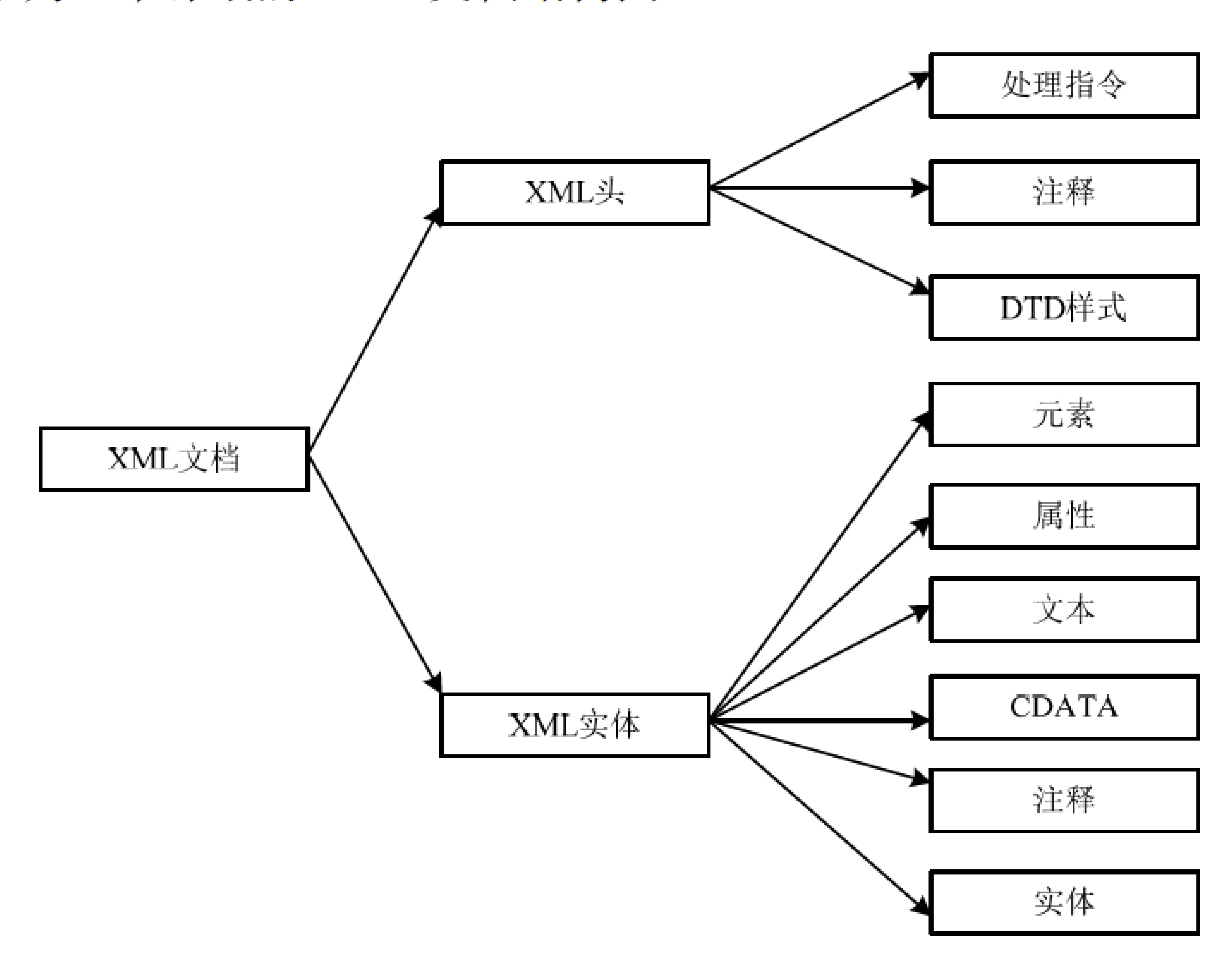


图 12-1　XML 文档结构图

由图 12-1 可知，在 XML 文档中包括 XML 头和 XML 实体，在添加 XML 头之后，才可以添加 XML 实体，下面详细介绍 XML 文档结构。

1. 声明 XML 文档

一个规范的 XML 文档通常以 XML 声明作为开始的一行，这一行代码会告诉解析器和浏览器，这个文件应该按照 XML 规则进行解析。XML 的声明以“<?xml”开始，以“?>”结束，其基本格式如下所示：

```
<?xml version="1.0" encoding="GB2312" standalone="yes/no"?>
```

在上面代码中使用了 3 个属性，下面分别介绍这 3 个属性的意义。

- version 属性：该属性不能被忽略且必须在属性列表的第一位，表示文档遵守 XML 规范的版本号，一般为 1.0。
- encoding 属性：表示文档的编码方式，默认是 UTF-8。如果 encoding 属性值设置为 GB2312，则 XML 文件必须使用 ANSI 编码保存。
- standalone 属性：该属性可选，表示该 XML 文档是否附带 DTD 文件，属性值为“yes”或“no”，如果属性值为 no，表示必须与外部 DTD 文件进行关联；如果是 yes，表示不需要与外部 DTD 文件进行关联。

2．处理指令

处理指令包含处理 XML 文档的文件和属性，以使 XML 解析器对 XML 文档做出正确的处理。处理指令格式如下所示：

```
<?xml-stylesheet type="text/css" href="css 文件位置"?><!--定义 CSS 文件-->
<?xml-stylesheet type="text/xsl" href="xsl 文件位置"?><!--定义 XSL 文件-->
```

3．DTD 样式

DTD 是一种保证 XML 文档格式正确的有效方法，可以通过比较 XML 文档和 DTD 文档来看 XML 是否符合规范，元素和标签使用是否正确。一个 DTD 文档包含元素的定义规则、元素间关系的定义规范、元素可使用的属性和可使用的实体。DTD 不仅可以确保 XML 文档符合既定的规则，而且还可以检验 XML 文档的有效性。

DTD 以“<!DOCTYPE”开始，以“>”结束。通常将开始和结束放在不同的行上，但是断行和多余的空格并不重要。文档类型的内部子集(即内部 DTD)定义在 XML 文档内，其语法格式如下所示：

```
<!DOCTYPE 根元素名[<!ELEMENT 根元素名(子元素列表)>]>
```

4．注释

XML 文档注释和 HTML 中的注释语法相同，非常容易理解和掌握。在 XML 文档添加注释的语法如下所示：

```
<!--注释内容-->
```



在注释内容中，不可包含“-->”字符，并且注释也不可放在声明前和标签中。

5．元素、属性和文本

元素、属性和文本是 XML 文档中最主要的内容，用于存储和表现数据。元素可以用来描述其所包含的数据，还可以包含属性名称和值，用于提供有关内容的其他信息。在 XML 文档中，只有一个根元素，其他元素在根元素内以树形分层结构排列，而且元素可以嵌套使用。XML 属性可以将一些额外的信息附加到元素上，文本表示元素之间的数据。这 3 个 XML 文档结构在 XML 文档中的表现形式如下所示：

```
<元素 属性='属性值'>文本</元素>
```

元素名和属性名可以包含字母、数字，以及其他一些合法的字符，但不能以数字或下划线开头，并且名称中不能包含空格或者任何形式的字符开头。对一些特殊符号，需要用实体符号代替，例如：“<”使用“<”代替、“>”使用“>”代替和“&”使用“&”代替等。属性的值必须包含在双引号之间，复杂的属性值需要通过子元素来实现，即元素可以实现嵌套。



不同于 HTML 会把多个相邻空格忽略为一个，XML 不处理空格，也就是会把空格保持原有的样式显示出来。

6. CDATA

CDATA 段以“<![CDATA[”开始，以“]]>”结束，两者中间的内容为文本内容。当 XML 元素被解析的时候，XML 元素内部的文本也会被解析。只有在 CDATA 段之内的文本才会被解析器当做一般文本直接显示。CDATA 格式如下所示：

```
<![CDATA[要直接显示的内容]]>
```

7. 实体

相当于 C 语言中的宏定义，可以先定义一个实体，然后通过实体名的形式来引用该实体。如上面提到过的特殊字符便是 XML 中预定义的实体，也可以通过 Unicode 或十六进制的形式来显示字符。

12.1.3 基础知识——命名空间

由于 XML 文档中使用的元素不固定，那么两个不同的 XML 文档使用同一个名字来描述不同类型的元素的情况就可能发生，发生这种情况时会导致命名冲突。在 XML 文档中，使用命名空间解决命名冲突的问题。

XML 命名空间被放置于某个元素的开始标签之中，并使用以下的语法：

```
xmlns:namespace-prefix="namespaceURL"
```

当一个命名空间被定义在某个元素的开始标签中时，所有带有相同前缀的子元素都会与同一个命名空间相关联。下面创建一个使用命名空间的 XML 文档，代码如下：

```
<?xml version="1.0" encoding="utf-8"?>
<tables>
    <table xmlns="http://www.w3.org/TR/html4/">
        <tr>
            <td>Ma Xiang Lin</td>
            <td>23</td>
        </tr>
    </table>
    <table xmlns="http://www.w3school.com.cn/furniture">
        <name>书桌</name>
        <width>120</width>
        <height>80</height>
    </table>
```

```
</tables>
```

在上面的代码中，创建了两个使用命名空间的 XML，并且通过使用命名空间解决了这两个 XML 文档之间的命名冲突问题。

12.1.4 实例描述

一个书店出售很多出版社的书籍，其中一个名为“O'Reilly”的出版社出版了《PHP Hacks》书，一个名为“Li'Juny”的出版社出版了《Podcasting Hacks》书。为了保存这样的图书信息数据，需要制作一个 XML 文件，那么，如何以出版社为条件分组呢？可以使用命名空间来划分出每个出版社的书籍。下面就来制作一个用于存储图书信息数据的 XML 文件吧！

12.1.5 实例应用

【例 12-1】 创建用于存储图书信息数据的 XML 文件

(1) 新建 XML 文件，命名为 13-1.xml。

(2) 编辑 13-1.xml 文件。该文件有一个“books”的根节点，在该根节点下存在多个子节点，每个出版社存有一个固定的命名空间地址。13-1.xml 文件的内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<!--指定 XML 文档的根节点 books-->
<books>
    <!--XML 文档的子节点 book-->
    <book xmlns="http://www.oreilly.com/">
      <!--书的作者-->
      <author>Jack Herrington</author>
      <!--书的名字-->
      <title>PHP Hacks</title>
      <!--出版社-->
      <publisher>O'Reilly</publisher>
    </book>
    <!--XML 文档的子节点 book-->
    <book xmlns="http://www.lijuny.com/">
        <!--书的作者-->
        <author>Jack Herrington</author>
        <!--书的名字-->
        <title>Podcasting Hacks</title>
        <!--出版社-->
        <publisher>Li'Juny</publisher>
    </book>
</books>
```

(3) 保存 13-1.xml 文件。

12.1.6　实例分析



源码解析：

在上述实例的 13-1.xml 文件中显示了一个图书列表，父标记<books>包含一组<book>标记，每个<book>标记又包含<author>、<title>和<publisher>标记，并且不同的出版社中的<book>标记中会有一个不同的命名空间地址，相同的出版社的命名空间地址是相同的，从而区分每个出版社所出版的书籍数据。

12.2　使用 PHP 生成记录新闻信息的 XML 文件

PHP 可以生成动态 HTML，也可以生成动态 XML。动态 XML 的一个典型应用是 RSS(丰富站点摘要，Rich Site Summary)，另一种解释为真正简易聚合(Really Simple Syndication)，一种用来同步新闻站点的文件格式。PHP 可以读取数据库或者 HTML 文件中的文章信息，生成一个基于这些信息的 XML 摘要文件，本节主要讲解如何运用 PHP 生成动态的 XML。



视频教学：光盘\videos\12\PHP 生成 XML.avi　　　　长度：5 分钟

12.2.1　基础知识——PHP 生成 XML

用 PHP 生成 XML 文档很简单，只需要用 header()函数把文档的 MIME 类型改为“text/xml”。为了避免<?xml ...?>声明被解释为一个 PHP 标签，需要编辑 php.ini 配置文件，将 short_open_tag 选项设为不启用，或者直接用 echo 把这一行打印出来：

```
<?php header('Content-Type:text/xml');
    echo "<?xml version='1.0' encoding='gb2312' ?>";
?>
```

使用 PHP 生成 XML 文档最简单的方式就是使用 PHP 语言生成字符串输出到页面中，生成 XML 文档。下面创建一个示例，代码如下：

```
<?php header('Content-Type:text/xml');
    echo "<?xml version='1.0' encoding='gb2312' ?>";
?>
<rss version="0.91">
    <channel>
        <?php
            $items=array(
                        array('title'=>'Web 成功开发路之 PHP',
                            'link'=>'http://www.itzcn.com',
                            'bookIntroduct'=>'本书主要从 PHP 基础讲解，然后到应用，
        特别对有一定 PHP 基础的人员使用，对于比较熟练的程序员也有技术上的帮助'),
                        array('title'=>'Java 从入门到精通',
```

```
                            'link'=>'http://www.itzcn.com',
                            'bookIntroduct'=>'本书主要讲解 Java 基础技术,以及 Java
                            的高级应用'
                        )
            );
            foreach ($items as $item)
            {
                echo "<item>\n";
                echo "<title>".$item['title']."</title>\n";
                echo "<link>".$item['link']."</link>\n";
                echo "<bookIntroduct>".$item['bookIntroduct']."</bookIntroduct>\n";
                echo "<language>en-us</language>\n";
                echo "</item>\n";
            }
        ?>
    </channel>
</rss>
```

从上面的代码可以看出，RSS 文档是一个包含有若干个 channel 元素的 XML 文档，每个 channel 包含有多个 item 元素，每个 item 元素又包含有一个 title、一个 link 和一个 bookIntroduct 元素。

执行上面的代码，PHP 自动生成 XML，生成的 XML 文档内容如下：

```
<?xml version='1.0' encoding='gb2312' ?>
<rss version="0.91">
    <channel>
        <item>
            <title>Web 成功开发路之 PHP</title>
            <link>http://www.itzcn.com</link>
            <bookIntroduct>本书主要从 PHP 基础讲解，然后到应用，特别对有一定 PHP 基础的
人员使用，对于比较熟练的程序员也有技术上的帮助</bookIntroduct>
            <language>en-us</language>
        </item>
        <item>
            <title>Java 从入门到精通</title>
            <link>http://www.itzcn.com</link>
            <bookIntroduct>本书主要讲解 Java 基础技术，以及 Java 的高级应用
            </bookIntroduct>
            <language>en-us</language>
        </item>
    </channel>
</rss>
```

12.2.2　实例描述

每天都会有各种各样的新闻报道，新闻就是广大群众欲知、应知而未知的重要事实，因此在当今世界新闻显得非常重要。通过新闻，人们可以知道国家、地区的最新动态等。下面使用 PHP 生成一个记录新闻信息的 XML 文件。

12.2.3　实例应用

【例 12-2】 用 PHP 生成记录新闻信息的 XML 文件

(1) 建 PHP 文件，命名为 13-2.php。

(2) 页面中定义新闻数据信息，声明一个名为“$data_array”的变量，存储数据，代码如下：

```
//数据
$data_array = array(
    array(
        'title' => '高校毕业生创业享税收减免',
        'content' => '新华社电 记者 3 日从教育部获悉，高校毕业生自主创业 3 年内按每户每
年 8000 元为限额依次扣减其当年实际应缴纳的营业税、城市维护建设税、教育费附加和个人所得税。
高校毕业生自主创业政策的具体内容是，对持《就业失业登记证》(注明“自主创业税收政策”或附着《高
校毕业生自主创业证》)毕业生从事个体经营(除建筑业、娱乐业以及销售不动产、转让土地使用权、广
告业、房屋中介、桑拿、按摩、网吧、氧吧外)的，在 3 年内按每户每年 8000 元为限额依次扣减其当年
实际应缴纳的营业税、城市维护建设税、教育费附加和个人所得税。 ',
        'pubdate' => '2011-01-05',
    ),
    array(
        'title' => '我国积极开展火星自主探测研究',
        'content' => '据中国空间技术研究院有关专家介绍，2011 年，我国将与俄罗斯合作共同
探测火星。与此同时，我国航天工业部门还先期启动了基于探月一二期技术的自主火星探测器研究和方
案设计工作，目前正在积极开展技术攻关，目标是瞄准 2013 年的发射“窗口”，利用我国“长征三号乙”
运载火箭发射。',
        'pubdate' => '2011-01-06',
    )
);
```

(3) 上面定义好的“title”元素中添加属性“size”，定义变量“$title_size”并赋值为 1，代码如下：

```
//向 title 元素中添加 size 属性，并赋值为 1
$title_size = 1;
```

(4) 编辑 XML 头部信息，并定义 XML 文档的根节点，代码如下：

```
//XML 头部信息
$xml = "<?xml version=\"1.0\" encoding=\"utf-8\"?>\n";
$xml .= "<article>\n";
```

(5) 基础工作已经做好了，最关键的就是编辑新闻信息数据，使用 PHP 调用上面已经定义好的数据自动生成 XML 文档。在这里使用函数的方式来生成新闻列表项，函数内容如下：

```
// 创建 XML 列表项
function create_item($title_data, $title_size, $content_data, $pubdate_data) {
    $item = "<item>\n";
    $item .= "<title size=\"" . $title_size . "\">" . $title_data . "</title>\n";
```

```
    $item .= "<content>" . $content_data . "</content>\n";
    $item .= " <pubdate>" . $pubdate_data . "</pubdate>\n";
    $item .= "</item>\n";
    return $item;
}
```

在这个函数中，需要传递 4 个参数，并生成一个名为“$item”的字符串，但并未在页面中输出。

(6) 循环遍历已经定义好的“$data_array”数组，并调用 create_item()函数，将数组中的元素传递到函数中，以字符串的形式输出到页面中，代码如下：

```
//循环遍历文章数据，生成字符串
foreach ($data_array as $data) {
    //调用下面的函数生成字符串
    $xml .= create_item($data['title'],
                $title_size,
                $data['content'],
                $data['pubdate']);
}
//最后以</article>结束
$xml .= "</article>\n";
//输出字符串
echo $xml;
```

(7) 保存 13-2.php 页面代码。

12.2.4 运行结果

执行 13-2.php 页面代码，在页面中输出一个完整的 XML 文档，内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<article>
    <item>
        <title size="1">高校毕业生创业享税收减免</title>
        <content>新华社电 记者 3 日从教育部获悉，高校毕业生自主创业 3 年内按每户每年 8000
元为限额依次扣减其当年实际应缴纳的营业税、城市维护建设税、教育费附加和个人所得税。高校毕业
生自主创业政策的具体内容是，对持《就业失业登记证》(注明“自主创业税收政策”或附着《高校毕业
生自主创业证》)毕业生从事个体经营(除建筑业、娱乐业以及销售不动产、转让土地使用权、广告业、
房屋中介、桑拿、按摩、网吧、氧吧外)的，在 3 年内按每户每年 8000 元为限额依次扣减其当年实际应
缴纳的营业税、城市维护建设税、教育费附加和个人所得税。 </content>
        <pubdate>2011-01-05</pubdate>
    </item>
    <item>
        <title size="1">我国积极开展火星自主探测研究</title>
        <content>据中国空间技术研究院有关专家介绍，2011 年，我国将与俄罗斯合作共同探测火
星。与此同时，我国航天工业部门还先期启动了基于探月一二期技术的自主火星探测器研究和方案设计
工作，目前正在积极开展技术攻关，目标是瞄准 2013 年的发射“窗口”，利用我国“长征三号乙”运载
火箭发射。</content>
```

```
        <pubdate>2011-01-06</pubdate>
    </item>
</article>
```

12.2.5　实例分析



源码解析：

上面案例使用了纯粹的 PHP 代码生成字符串，并把这个字符串写入到文件中，这是最原始的生成 XML 的方法，也是最简单的。这里只讲述这种自动生成 XML 文档的方法。

在上面的案例中向“title”元素中添加了一个名为“size”的属性，声明一个名为“$title_size”的变量。在 create_item()函数中的“$item”字符串中，向“<title>”标签里添加了一个“size”属性，即“<title size=$title_size”。

12.3　PHP 解析方法比较

在 PHP 中，可以使用正则表达式和字符串函数解析 XML 文档；可以使用 DOM 库对 XML 文档进行解析；可以使用基于 Expat C 库的事件驱动型解析器(即 SAX 解析器)解析 XML 文档，还可以使用适合于解析简单 XML 文件的解析器 SimpleXML，本节主要对这些解析 XML 文档的方法进行简单介绍和对比。



视频教学：光盘\videos\12\XML 解析方式.avi　　　　长度：5 分钟

12.3.1　基础知识——PHP 解析方法比较

PHP 主要提供了 4 种解析 XML 文档的解析器，即使用正则表达式和字符串函数解析 XML 文档、使用 DOM 库对 XML 文档进行解析、使用 SAX 解析器解析 XML 文档和使用 SimpleXML 解析器解析 XML 文档，下面就对这 4 种解析器进行简单介绍，比较它们之间的异同点。

1．使用正则表达式和字符串函数解析 XML 文档

使用正则表达式读取 XML 文档，必须确保 XML 的格式良好，这样才可以顺利地读取，但是有些格式正确的 XML 可能与正则表达式不匹配，所以一般不建议使用正则表达式读取 XML。但是它是兼容性最好的方式，因为正则表达式函数总是可用。不要用正则表达式直接读取来自用户的 XML，因为无法控制这类 XML 的格式和结构。

2．使用 DOM 库对 XML 文档进行解析

使用 DOM(Document Object Model 的缩写)接口处理 XML 文档，需要把整个 XML 文件加载到内存中去，并以一棵节点树的形式存在。当加载完成后，可以对节点树中的每一个节点进行操作。换句话说，通过 DOM 树，应用程序可以对 XML 文档进行随机访问。这种访问方式给应用程序的开发带来了很大的灵活性，可以任意地控制整个 XML 文档中的内容，从而执行

添加、删除、修改等操作。例如，下面代码所示为一个简单的 XML 文档，以它为例进行解析。

```
<?xml version="1.0" encoding="utf-8"?>
<myClass>
    <unit>
        <id>1</id>
        <name>创建 XML 文件</name>
        <address>http://www.itzcn.com</address>
    </unit>
</myClass>
```

上述就是一个由节点组成的 DOM 树结构。其中，myClass 是文档的根节点，在根元素里面包含一个 unit 子节点，而 unit 元素又包含了 3 个子节点：id 节点、name 节点和 address 节点，并且都包含了响应的数据信息。在 DOM 中，每一个元素都会成为一个节点，属性也是一个节点，标记间的数据也是一个节点，可以清晰地表达出文档中各节点之间的关系，图 12-2 所示为上述的 XML 文档结构图。

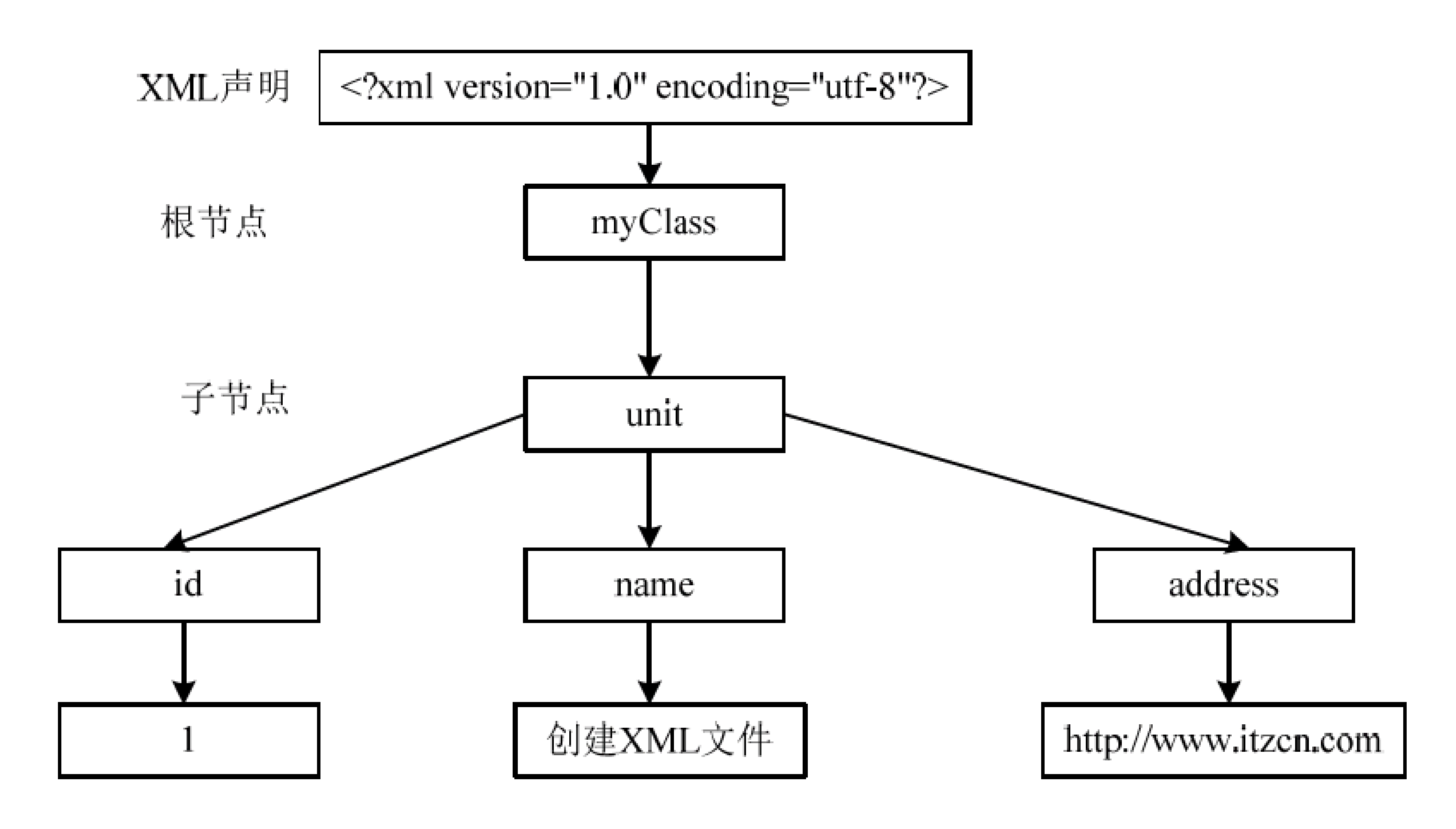


图 12-2　DOM 解析 XML 流程

由于 DOM 解析器把整个 XML 文档转化成 DOM 树放在了内存中，因此，当 XML 文档比较大或者文档结构比较复杂时，对内存的要求就比较高。而且，对于结构复杂的树的遍历也是一项比较耗时的操作。所以，DOM 解析器对硬件性能的要求比较高，实现效率并不十分理想。由于 DOM 解析器的树结构与 XML 文档结构相吻合，所以通过 DOM 树机制很容易实现随机访问。

3．使用 SAX 解析器解析 XML 文档

解析 XML 的另一种方法是使用 XML Simple API(SAX)解析器。SAX 解析器运行在回调模型上，每次打开或关闭一个标签时，或者每次解析器看到文本时，就用节点或文本的信息回调用户定义的函数。SAX 解析器在解析 XML 文档时，不需要验证 XML 文档，就可以找出当前的 XML 标签及其包含的内容，但是却不能确定它们是否为正确文档结构中的标签。图 12-3 中给出了使用 SAX 解析 XML 时的流程。

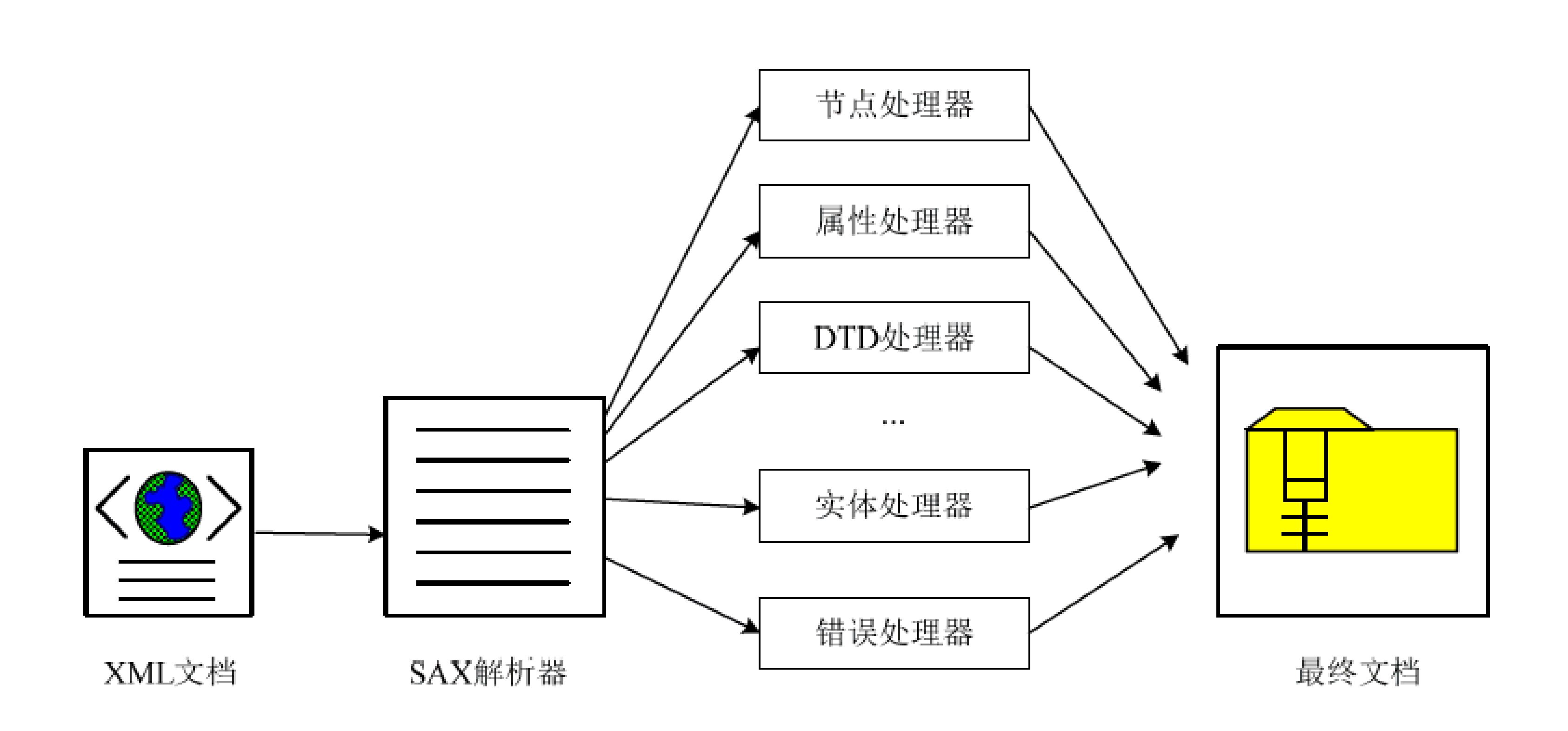


图 12-3　SAX 解析 XML 流程

SAX 解析器的优点在于，它是真正轻量级的解析器，不会在内存中长期保存内容，所以可以用于非常大的文件。缺点是编写 SAX 解析器回调非常麻烦。同 DOM 分析器相比，SAX 解析器对 XML 文档的处理缺乏一定的灵活性，然而对于那些只需要访问 XML 文档中的数据而不对文档进行更改的应用程序来说，SAX 解析器的效率则更高。

4．使用 SimpleXML 解析器解析 XML 文档

如果需要解析一些简单的 XML 文档，推荐使用 SimpleXML 解析器。SimpleXML 解析器可以非常容易地操作 XML 文档，但是 SimpleXML 无法像 DOM 那样扩展生成 XML 文档树，也不能像 SAX 那样灵活与高效。

12.3.2　实例描述

XML 用于存储数据，因此可以编写 PHP 读取 XML 数据内容并展示在页面中，这是经常需要做的一件事。在学生信息管理系统中，需要从 XML 文档中读取学生信息，并在页面中以列表的形式显示，而解析 XML 文档有多种方式，下面就用正则表达式的方式来读取 XML 文档数据，做一个学生信息管理页面。

12.3.3　实例应用

【例 12-3】 制作学生信息管理页面

(1) 新建 XML 文件，命名为 student.xml。

(2) 在 student.xml 文档中编辑学生信息数据，设置文档的根节点为“list”，在根节点中有多个“student”子节点。文档内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<list>
    <!-- 设置第一条数据-->
    <student>
        <!-- 学生姓名-->
        <name>王丽丽</name>
        <!-- 学生班级-->
```

```
        <class> 2 班</class>
        <!-- 出生日期-->
        <birth> 1988-02-21</birth>
        <!-- 星座-->
        <constell>双鱼座</constell>
        <!-- 联系电话-->
        <mobile>12568487895</mobile>
    </student>
    <!-- 设置第二条数据-->
    <student>
        <!-- 学生姓名-->
        <name>张赞红</name>
        <!-- 班级-->
        <class>2 班</class>
        <!-- 出生日期-->
        <birth>1988-01-24</birth>
        <!-- 星座-->
        <constell> 双鱼座</constell>
        <!-- 联系电话-->
        <mobile> 13658478957</mobile>
    </student>
    <!-- 设置第三行数据-->
    <student>
        <!-- 学生姓名-->
        <name>张晓强</name>
        <!-- 学生所在的班级-->
        <class>2 班</class>
        <!-- 出生日期-->
        <birth> 1988-06-21</birth>
        <!-- 星座-->
        <constell>双子座</constell>
        <!-- 联系电话-->
        <mobile>15748789548</mobile>
    </student>
    <!-- 第四条数据-->
    <student>
        <!-- 学生姓名-->
        <name>彭玉</name>
        <!-- 学生所在的班级-->
        <class>1 班</class>
        <!-- 出生日期-->
        <birth>1988-06-13</birth>
        <!-- 星座-->
        <constell>双子座</constell>
        <!-- 联系电话-->
        <mobile>13578487598</mobile>
    </student>
</list>
```

(3) 保存 student.xml 文档。

(4) 新建 PHP 文件，命名为 13-3.php。

(5) 编辑 13-3.php 页面，在页面中使用 fopen()函数打开上面编辑好的 student.xml 文件，并使用 fread()函数读取文件，然后使用 preg_match_all()函数查找匹配的内容，最后以"echo"输出在页面中，代码如下：

```
<?php
    //定义变量，用于存储读取到的内容
    $xml = "";
    //打开 student.xml 文件
    $f = fopen( 'student.xml', 'r' );
    //读取文件内容，将获取到的信息赋值给$xml 变量
    while( $data = fread( $f, 4096 ) )
    {
        $xml .= $data;
    }
    //关闭文件
    fclose( $f );
    //使用 preg_match_all()函数全文匹配，即在$xml 中查找满足正则表达式的都找出来并赋值
给$bookblocks
    preg_match_all( "/\<student\>(.*?)\<\/student\>/s", $xml, $bookblocks );
    //输出表格的开始
    echo  "<TABLE  cellSpacing=0  cellPadding=2  width='95%'  align=center
border=0>";
    //循环遍历查找到的所有项
    foreach( $bookblocks[1] as $block )
    {
        //在每一项中查找满足正则表达式的内容，并赋值给$name 变量
        preg_match_all( "/\<name\>(.*?)\<\/name\>/",$block, $name );
        //在每一项中查找满足正则表达式的内容，并赋值给$class 变量
        preg_match_all( "/\<class\>(.*?)\<\/class\>/",$block, $class );
        //在每一项中查找满足正则表达式的内容，并赋值给$birth 变量
        preg_match_all( "/\<birth\>(.*?)\<\/birth\>/",$block, $birth );
        //在每一项中查找满足正则表达式的内容，并赋值给$constell 变量
        preg_match_all(     "/\<constell\>(.*?)\<\/constell\>/",     $block,
$constell );
        //在每一项中查找满足正则表达式的内容，并赋值给$mobile 变量
        preg_match_all( "/\<mobile\>(.*?)\<\/mobile\>/",$block, $mobile );
        //输出表格中的行
        echo "<TR><TD style='COLOR: #880000'>";
        //输出单元格中的内容，将获取到的学生信息输出到页面中
        echo  "姓名："."$name[1][0]."  班级："."$class[1][0]." 
 出生日期："."$birth[1][0]."  星座："."$constell[1][0]." 
 联系电话："."$mobile[1][0]."</TD></TR>";
    }
    //输出表格的结尾
    echo "</TABLE>";
?>
```

(6) 新建 header.htm 页面和 menu.htm 页面，分别作为框架的顶部和左边页面。

(7) 新建 index.htm 页面，在页面中使用框架，将 header.htm 页面、menu.htm 页面和 13-3.php 页面整合。

12.3.4 运行结果

在浏览器中运行 index.htm 页面，在页面中输出学生信息列表，如图 12-4 所示。



图 12-4 学生信息列表页面

12.3.5 实例分析



源码解析：

在上面实例中的 13-3.php 页面中，开始使用正则表达式将文件读取到一个大的字符串中，然后使用 preg_match_all()函数读取每个学生项目，最后使用 foreach 循环，在每条学生信息块间循环，并提取出学生姓名、班级、出生日期、星座和联系电话。

使用正则表达式代码读取 XML 的问题是，它并没先进行检查，确保 XML 的格式良好，这意味着在读取之前，无法知道 XML 格式是否良好，而且，有些格式正确的 XML 可能与正则表达式不匹配，因此会导致不必要的麻烦。

12.4 使用 SAX 解析 XML 文档中的书籍信息数据并显示

SAX 解析器是基于 Expat C 库的事件驱动型解析器，该 PHP 解析器也是最常用的解析器之一。由于 SAX 解析器基于事件，所以在解析时不要求验证 XML 文档，可以直接找出当前的 XML 标签及其包含的内容，但是却不能确定它们是否为正确文档结构中的正确标签，不过在实际中，该问题可以不考虑。

当 SAX 解析器阅读文档时，将针对各种事件调用不同的处理程序。本节将主要围绕 SAX

解析器讨论可以提供的处理程序，以及设置这些处理程序的函数和触发调用处理程序的事件。

视频教学：光盘\videos\12\SAX 方式操作 XML.avi　　长度：6 分钟

12.4.1　基础知识——创建解析器

在 PHP 中使用 xml_parser_create()函数创建 XML 解析器，然后设置该解析器的处理器和选项。xml_parser_create()函数建立一个新的 XML 解析器并返回可被其他 XML 函数使用的资源句柄。该函数语法格式如下所示：

```
resource xml_parser_create ( [string encoding] )
```

由 xml_parser_create()函数语法格式可知，该函数有一个可选参数，表示指定将要解析的文本编码，例如 ISO-8859-1、US-ASCII 或 UTF-8。创建解析器实例代码：

```
$parser=xml_parser_create("UTF-8");
```



在 PHP 5.0.0 和 5.0.1 中，默认输出的字符编码是 ISO-8859-1，而 PHP 5.0.2 及以上版本是 UTF-8。解析器支持的编码有 ISO-8859-1、UTF-8 和 US-ASCII。

创建解析器之后，再使用 xml_parse()函数解析 XML 文档，并在此时触发元素处理器、字符数据处理器和默认处理器等。数据处理完毕或者解析器返回一个错误，调用 xml_parser_free()函数释放解析器。xml_parse()函数语法格式如下所示：

```
int xml_parse ( resource parser, string data [, bool is_final] )
```

xml_parse()函数有 3 个参数：parser 参数表示一个指向将要使用的 XML 解析器的指针；data 参数表示需要解析的数据集，可以多次对新的数据调用 xml_parse()函数来分段解析一个文档，只需要在解析最后一段数据时将 is_final 参数设置为 true；可选的 is_final 参数被设为 true，则 data 为当前解析中最后一段数据。如果解析器成功解析 XML，xml_parse()函数将返回 true，否则返回 false。还可以使用 xml_get_error_code()函数获取错误的标识性代码，这些错误代码可能是以下之一：

```
XML_ERROR_NONE
XML_ERROR_NO_MEMORY
XML_ERROR_SYNTAX
XML_ERROR_NO_ELEMENTS
XML_ERROR_INVALID_TOKEN
XML_ERROR_UNCLOSED_TOKEN
XML_ERROR_PARTIAL_CHAR
XML_ERROR_TAG_MISMATCH
XML_ERROR_DUPLICATE_ATTRIBUTE
XML_ERROR_JUNK_AFTER_DOC_ELEMENT
XML_ERROR_PARAM_ENTITY_REF
XML_ERROR_UNDEFINED_ENTITY
XML_ERROR_RECURSIVE_ENTITY_REF
XML_ERROR_ASYNC_ENTITY
XML_ERROR_BAD_CHAR_REF
```

```
XML_ERROR_BINARY_ENTITY_REF
XML_ERROR_ATTRIBUTE_EXTERNAL_ENTITY_REF
XML_ERROR_MISPLACED_XML_PI
XML_ERROR_UNKNOWN_ENCODING
XML_ERROR_INCORRECT_ENCODING
XML_ERROR_UNCLOSED_CDATA_SECTION
XML_ERROR_EXTERNAL_ENTITY_HANDLING
```



获取标识性代码之后，再使用 xml_error_string()函数将错误代码转换成字符串。

12.4.2　基础知识——元素处理器

由于 SAX 解析器是基于 Expat C 库的事件驱动型解析器，因此当解析器遇到一个元素的开始标签或结束标签时，SAX 解析器会调用起始或结束元素处理器。PHP 通过 xml_set_element_handler()函数来设置该处理器，语法格式如下：

```
bool xml_set_element_handler ( resource parser, callback start_element_handler,
callback end_element_handler )
```

参数 parser 表示要使用的 XML 解析器，start_element_handler 参数和 end_element_handler 参数表示处理器函数的名称，并且这两个函数都是按引用传递。当 XML 解析器遇到元素的起始标签时，会调用 start_element_handler 元素处理函数，该函数必须有 3 个参数，其语法格式如下所示：

```
my_start_element_handler(parser , element , attributes)
```



起始元素处理器函数按引用传递给 my_start_element_handler()函数。

其中，my_start_element_handler()函数的参数中：

- parser，表示调用处理程序的 XML 解析器引用。
- element，表示起始元素的名称。
- attributes，表示解析器遇到元素的属性数组。

下面创建一个包含起始元素处理器的实例，该实例代码如下：

```
<?php
function start($parser,$element_name,$element_attrs){
    //echo $element_name;
    switch($element_name)//使用 switch 语句遍历起始标签
    {
        case "LIST":
            echo "<h4>学生信息列表</h4>";
            break;
        case "STUDENT":
            echo "<br/><span style='font-size:18px; color:red'>--------学生信
息----------</span><br/>";//当起始标签为 student 时，输出学生信息
            break;
```

```
        case "NAME":
            echo "学生姓名：";           //当起始标签为name时，输出“学生姓名”
            break;
        case "BIRTH":
            echo "出生日：期";           //当起始标签为birth时，输出“出生日期”
            break;
        case "CLASS":
            echo "学生所在的班级：";     //当起始标签为class时，输出“学生所在的班级”
            break;
        case "CONSTELL":
            echo "星座：";               //当起始标签为constell时，输出“星座”
            break;
        case "MOBILE":
            echo "联系电话：";
            break;
    }
}
?>
```

当 XML 解析器遇到元素的结束标签时，会调用 end_element_handler 参数所代表的处理函数。结束标签元素处理函数语法格式如下所示：

```
my_end_element_handler(parser , element)
```

end_element_handler 参数所代表的处理函数有两个参数：parser 表示调用处理程序的 XML 解析器的引用；element 参数表示结束标签的名称。下面创建一个结束标签元素处理函数，代码如下：

```
function stop($parser,$element_name)
{
    echo " ";
}
```

12.4.3　基础知识——字符数据处理器

元素之间的所有文本(如字符数据、XML 术语中的 CDATA)由字符数据处理器处理。在 PHP 中，通过 xml_set_character_data_handler()函数设置的处理器会在遇到第一个字符数据块时被调用，函数语法格式如下所示：

```
bool xml_set_character_data_handler ( resource parser, callback handler )
```

字符数据处理器有两个参数：parser 参数表示指向 XML 解析器指定字符数据处理函数；handler 参数表示事件处理器使用的函数。下面创建一个事件处理器使用的函数，代码如下：

```
<?php
    function char($parser,$data)
    {
        echo $data;      //输出节点文本
    }
```

```
?>
```

在前面两节创建了 XML 解析器、元素处理器和字符数据处理器，此时，就可以打开 XML 文档来读取，然后使用解析器、元素处理器和字符数据处理器解析 XML 文档。SAX 解析 XML 文档实例代码如下所示：

```
    xml_set_element_handler($parser,"start","stop");    //元素处理器
    xml_set_character_data_handler($parser,"char");     //字符数据处理器
    $fp=fopen("student.xml","r");
    while ($data=fread($fp,4096))        //循环读取 XML 文档中的数据
    {
        xml_parse($parser,$data,feof($fp)) or die(sprintf("XML Error: %s at
line  %d",xml_error_string(xml_get_error_code($parser)),xml_get_current_line_number
($parser)));                            //解析 XML 文档
    }
    xml_parser_free($parser);;
```

本实例使用的 XML 文档就是上一节案例中用到的 student.xml 文件，这里不再重复。

在上面的代码中，首先创建了元素处理器和字符数据处理器，然后打开 XML 文档。接下来使用 while 循环读取 XML 文档中的数据，然后使用 xml_parse()函数解析 XML 文档，当遇到元素标签时，会调用元素处理器函数，当遇到元素之间的数据时，会调用字符数据处理器函数。最后使用 xml_parser_free()函数释放 XML 解析器。运行程序，页面效果如图 12-5 所示。

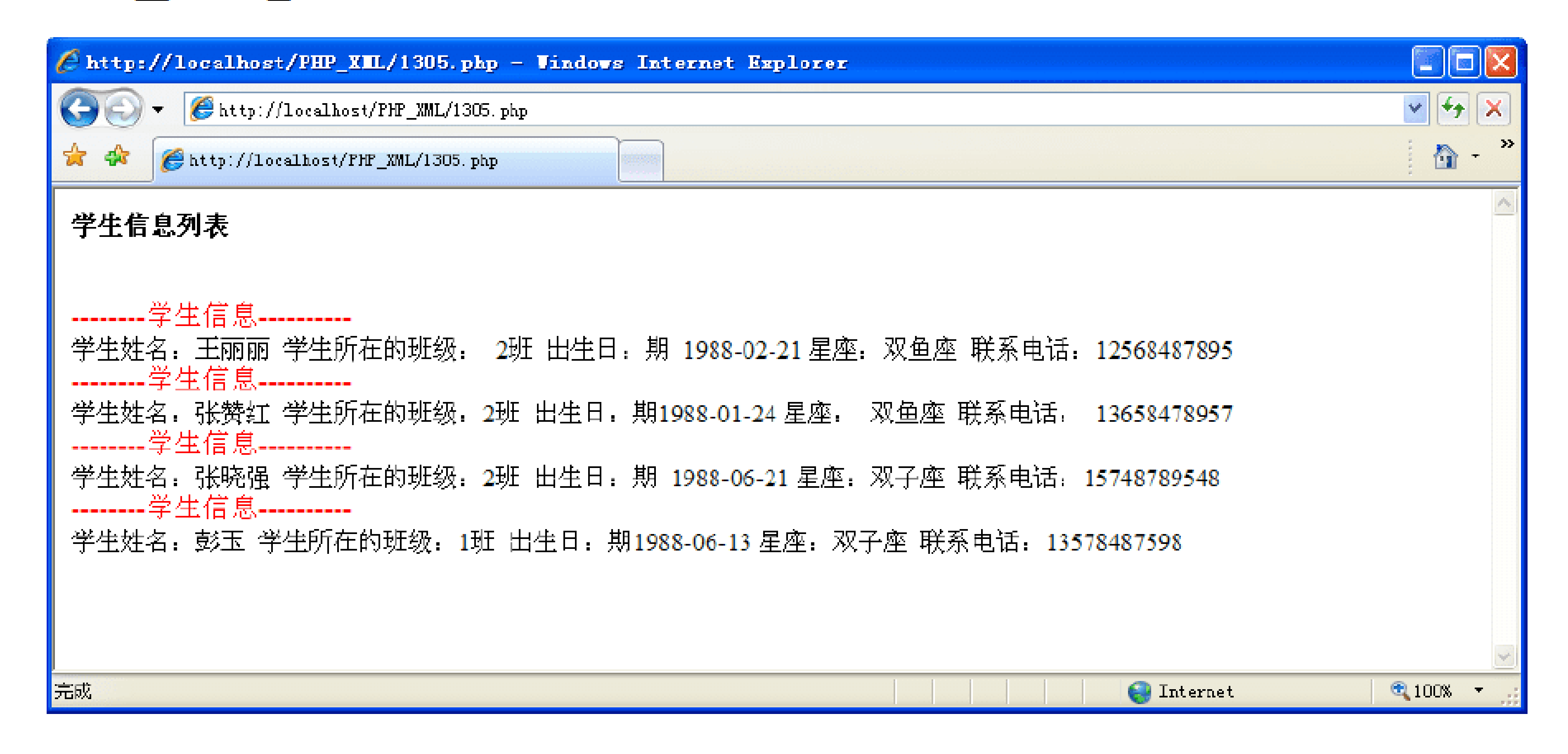


图 12-5　解析 XML 文档页面效果图



xml_parse()函数将对 XML 文档进行逐行解析。

12.4.4　基础知识——指令处理器

在 XML 中处理指令(PI，Processiong Instruction)包含在“<?”和“?>”分隔符中，用于将脚本或其他代码嵌入到文档中。XML 解析器在遇到处理指令时即调用相应的指令处理器。PHP 使用 xml_set_processing_instruction_handler()函数设置处理器，语法格式如下所示：

```
bool   xml_set_processing_instruction_handler   (   resource   parser,   callback
handler )
```

该函数允许传递两个参数，第一个参数表示调用处理程序的 XML 解析器的引用，和前面所讲述的几个函数相同；第二个参数表示要调用的处理器函数。有 handler 指定的函数必须有三个参数：触发处理器的解析器的引用、目标名称和处理指令。语法格式如下所示：

```
my_processiong_instruction_handler(parser,target,instructions)
```

当解析文档时，用 eval()函数执行 PHP 代码，如下面的代码。当然这必须依赖正在处理的包含 eval()的文档，因为 eval()函数将运用所包含的任何代码——即使该代码会删除文件或者把密码发送给黑客。

```
function procession_instruction($inParser,$inTarget,$inCode)
{
        if($inTarget='php')
        {
            eval($inCode);
        }
}
```

下面创建一个实例，具体讲述处理指令函数的应用。创建 XML 文件，命名为 stu.xml，文件内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<students>
    <student>
        <!-- 学生姓名-->
        <name>张赞红</name>
        <!-- 班级-->
        <class>2 班</class>
        <!-- 出生日期-->
        <birth>1988-01-24</birth>
        <!-- 星座-->
        <constell> 双鱼座</constell>
        <!-- 联系电话-->
        <mobile> 13658478957</mobile>
    </student>
</students>
```

编辑处理指令函数解析 XML 文件，代码如下：

```
//省略了前面已经列出过的函数
    function procession_instruction($inParser,$inTarget,$inCode)
    {
        if($inTarget='php')
        {
            eval($inCode);
            echo "<br/>";
        }
    }
```

```
    xml_set_processing_instruction_handler($parser,"procession_instruction")
;//指令处理器
    xml_set_element_handler($parser,"start","stop");    //元素处理器
    xml_set_character_data_handler($parser,"char");     //字符数据处理器
    $fp=fopen("stu.xml","r");
    while ($data=fread($fp,4096))   //循环读取 XML 文档中的数据
    {
        xml_parse($parser,$data,feof($fp)) or die(sprintf("XML Error: %s at
line  %d",xml_error_string(xml_get_error_code($parser)),xml_get_current_line_number
($parser)));                        //解析 XML 文档
    }
    xml_parser_free($parser);
```

由于 PHP 代码本身就可被看成是一种遵循 XML 格式的处理指令，所以，可以将 PHP 代码嵌入到 XML 文档中，然后在处理器函数中处理 PHP 程序。在上面的代码中，procession_instruction()处理器函数使用了 eval()函数执行 XML 文档中的 PHP 程序。运行程序，页面效果如图 12-6 所示。

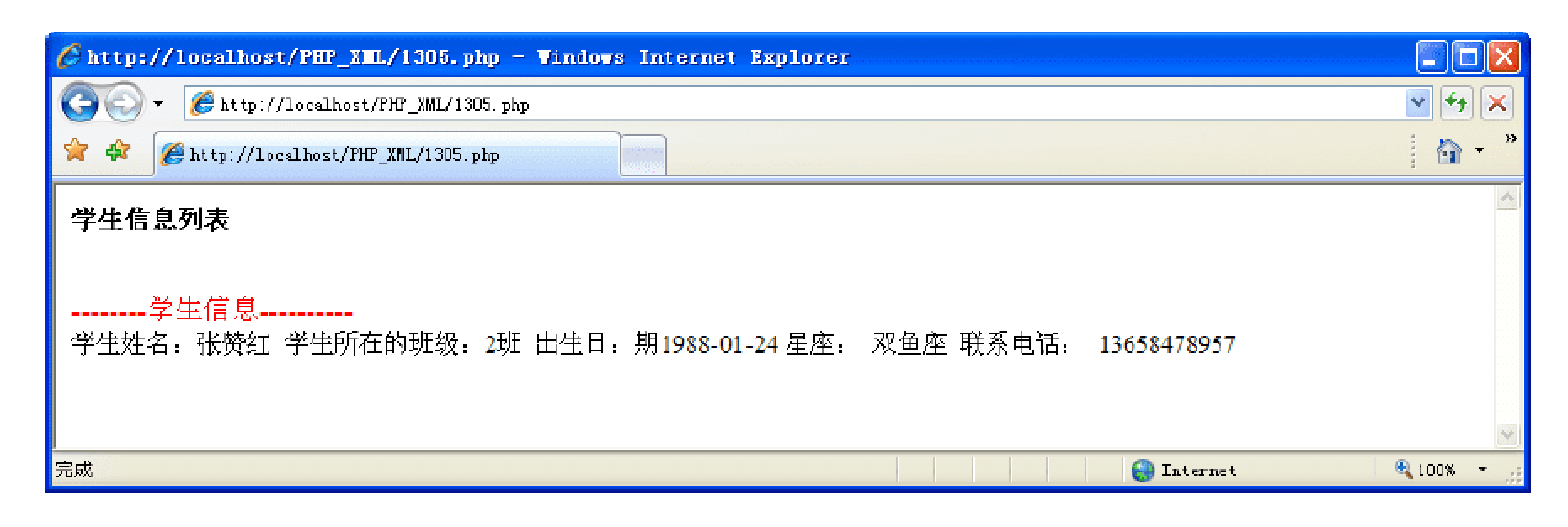


图 12-6　处理指令实例页面效果图

12.4.5　基础知识——实体处理器

XML 中的实体是占位符(placeholder)，XML 提供了 5 种标准实体(&、>、<、"和')，并且 XML 文档还可以自定义实体。大多数实体定义不会触发事件，在调用其他处理器前，XML 解析器会对文档中的大多数实体进行扩展。

PHP 的 XML 库对外部实体(external entity)和未解析实体(unparsed entity)提供特别支持。外部实体的替换文本(replacement text)是由文件名或者 URL 表示，而不是在 XML 文件中显示的指定外部实体的替换文本。程序员可以定义一个处理器，并在字符数据中出现外部实体时调用，还可以解决是否解析外部实体文件或 URL 的内容。未解析实体必须有一个符号声明。程序员可以定义未解析实体和符号声明，并且未解析实体将在调用字符数据处理器前被删除。下面详细介绍这两种实体。

1．外部实体

外部实体允许 XML 文档包含其他 XML 文档。一个外部实体引用的处理器首先打开被引用的文件，然后解析该文件并且将结果包含在当前文档中。可用 xml_set_external_entity_ref_handler()函数来设置处理器。如果处理器被成功建立，该函数返回 true，否则返回 false。xml_set_

external_entity_ref_handler()函数的语法格式如下所示：

```
bool xml_set_external_entity_ref_handler ( resource parser, callback handler )
```

该函数允许传递两个参数：parser 参数表示要使用的 XML 解析器；handler 参数表示当解析器找到外部实体时被调用的函数。

handler 参数指定的外部实体处理器函数必须有 5 个参数，语法格式如下：

```
my_external_entity_ref_handler(parser , name , base , system_id , public_id)
```

这 5 个参数都是必须的，它们所表示的意义如下。

- parser：规定一个变量，包含调用处理器的 XML 解析器。
- name：规定包含外部实体名称的变量。
- base：规定一个变量，包含解析外部实体的系统标识符(system_id)的基础。该参数通常被设置为空字符串。
- system_id：规定包含外部实体的系统标识符的变量。
- public_id：规定包含外部实体的公共标识符的变量。

如果外部实体处理器返回一个 false 值或者没有返回任何值，XML 解析器将停止并报告一个 XML_ERROR_EXTERNAL_ENTITY_HANDLING 错误；如果返回 true，则继续解析。下面创建一个实例，该实例所用到的 XML 文档内容如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<article>
    <author>马向林&amp;</author>
    <from>http://www.itzcn.com&amp;</from>
    <head>SAX 解析 XML 之外部实体</head>
    <body>&amp;,&gt;,&lt;,&quot;和 &apos;</body>
</article>
```

该实例主要介绍如何解析外部引用的 XML 文档，它定义了两个函数，分别用来创建 XML 解析器和进行解析工作，可以用它们来解析顶层文档和由外部引用包含的任何文档，外部实体引用处理器只是简单地确定哪些文件要传递给这些函数。使用外部实体解析 XML 文档的代码如下所示：

```
<?php
    //创建一个解析器
    $parser=xml_parser_create();
    //字符处理器函数
    function char($parser,$data)
    {
        echo $data;
    }
    //实体处理器函数
    function external_entity_handler($parser,$name,$base,$sysId,$pubId)
    {
        echo "$name";
        echo "$sysId";
        echo "$pubId";
```

```
    }
    //字符数据处理器
    xml_set_character_data_handler($parser,"char");
    //实体处理器
    xml_set_external_entity_ref_handler($parser,"external_entity_handler");
    //打开 article.xml 文件
    $fp=fopen("article.xml","r");
    //循环读取 XML 文件数据
    while ($data=fread($fp,4096)) {
        //解析 XML 文档，当出现错误时，返回错误
        xml_parse($parser,$data,feof($fp)) or die(sprintf("XML Error: %s at
line  %d",xml_error_string(xml_get_error_code($parser)),xml_get_current_line_number
($parser)));
    }
    //释放解析器
    xml_parser_free($parser);
?>
```

在浏览器中查看运行结果，在页面中输出：

```
马向林& http://www.itzcn.com& SAX 解析 XML 之外部实体 &,>,<,"和 '
```

2. 未解析实体

未解析实体的声明要带有一个符号声明，如下所示：

```
<!DOCTYPE doc[
            <!NOTATION jpeg SYSTEM "image/jpeg">
            <!ENTITY logo SYSTEM "php-tiny.jpg" NDATA jpeg>
]>
```

当解析器在 XML 文档中遇到符号声明时调用 xml_set_notation_decl_handler()函数，使用该函数可以注册一个符号声明处理器。该函数的语法格式如下：

```
bool xml_set_notation_decl_handler ( resource parser, callback handler )
```

xml_set_notation_decl_handler()函数有两个参数：parser 参数规定要使用的 XML 解析器；handler 参数规定当解析器找到符号声明时被调用的函数，这两个参数都是必需的。如果处理器被成功建立则返回 true，否则返回 false。handler 参数所表示的处理器函数需要 5 个参数，函数的语法格式如下：

```
my_notation_handler( parser , notation , base , system , public)
```

其中，base 参数表示分解符号标识符的基本 URL，可以设置 system 标识符或 public 标识符，但是不能两者同时设置。

12.4.6 基础知识——默认处理器

只要解析器在 XML 文档中找到数据，就可以调用 xml_set_default_handler()函数解析 XML 文档。xml_set_default_handler()函数为 XML 解析器建立默认的数据处理器，如果处理器被成

功创建则返回 true，否则返回 false。该函数语法格式如下：

```
bool xml_set_default_handler ( resource parser, callback handler )
```

xml_set_default_handler()函数有两个参数：parser 参数表示要使用的 XML 解析器；handler 参数表示作为事件处理器使用的函数。下面用该函数创建一个实例，该实例所使用的 XML 文件代码如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<note>
    <to>George</to>
    <from>John</from>
    <heading>Reminder</heading>
    <body>Don't forget the meeting!</body>
</note>
```

接着创建 1307.php 页面，在页面中编辑默认处理器函数，使用默认处理器解析 XML 文档。PHP 代码如下：

```
<?php
    //新建解析器
    $parser=xml_parser_create();
    //默认处理器函数
    function mydefault($parser,$data)
    {
      echo $data;
    }
    //默认处理器
    xml_set_default_handler($parser,"mydefault");
    //打开 XML 文档
    $fp=fopen("note.xml","r");
    //循环读取 XML 文档
    while ($data=fread($fp,4096))
    {
      //解析 XML 文档
      xml_parse($parser,$data,feof($fp)) or
      die (sprintf("XML Error: %s at line %d",
      xml_error_string(xml_get_error_code($parser)),
      xml_get_current_line_number($parser)));
    }
    xml_parser_free($parser);
?>
```

在浏览器中运行 1307.php 页面，在页面中输出结果为：

```
George John Reminder Don't forget the meeting!
```

在浏览器中查看该页面的源代码，内容如下所示：

```
<note>
    <to>George</to>
```

```
    <from>John</from>
    <heading>Reminder</heading>
    <body>Don't forget the meeting!</body>
</note>
```

12.4.7 实例描述

对于一个销售书籍的商店来说，图书管理系统是不可或缺的。通过图书管理系统可以对书籍进行增、删、改的操作，当然查询是最基础的一种操作，在对书籍进行增、删、改之前需要把本商店中的所有书籍信息查询到页面中，这里采用 XML 文档来存储书籍信息，因此，这里的查询也是从 XML 文档中读取书籍信息数据的。

12.4.8 实例应用

【例 12-4】 使用 SAX 解析 XML 文档中的书籍信息数据并显示

(1) 新建 XML 文件，命名为 book.xml。

(2) 编辑 book.xml 文件，编辑书籍信息数据，内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
    <book>
        <!--作者-->
        <author>马向林</author>
        <!--书名-->
        <name>《J2EE 开发实战》</name>
        <!--出版日期-->
        <time>2011-01-01</time>
        <!--出版社-->
        <publisher>电子信息出版社</publisher>
    </book>
    <book>
        <author>马向林</author>
        <name>《PHP 高级应用》</name>
        <time>2011-02-01</time>
        <publisher>方立出版社</publisher>
    </book>
</books>
```

(3) 保存 book.xml 文件。

(4) 新建 PHP 文件，命名为 13-4.php。

(5) 编辑 13-4.php，在页面中声明两个全局变量，代码如下：

```
    //定义 XML 文档中节点组成的数组
    $g_books = array();
    //存储 XML 文档中获取的元素开始名称
    $g_elem = null;
```

(6) 定义起始元素处理器、结束元素处理器和字符数据处理器三个函数，代码如下：

```
    //定义起始元素处理器函数
    function startElement( $parser, $name, $attrs )
    {
        //使用 global()函数将全局变量$g_books 和$g_elem 引入函数内部
        global $g_books, $g_elem;
        //如果起始元素的名称为“BOOK”，则实例化数组
        if ( $name == 'BOOK' )
            $g_books []= array();
        //将开始元素名称赋值给$g_elem 变量
        $g_elem = $name;
    }
    //定义结束元素处理器函数
    function endElement( $parser, $name )
    {
        //使用 global()函数将全局变量引入结束元素处理器函数内部
        global $g_elem;
        //清空元素变量
        $g_elem = null;
    }
    //定义字符数据处理器函数
    function textData( $parser, $text )
    {
        //使用 global()函数将全局变量$g_books 和$g_elem 引入函数内部
        global $g_books, $g_elem;
        //如果读取的 XML 开始元素名称为 AUTHOR、NAME、TIME 和 PUBLISHER，保存至数组中
        if ( $g_elem == 'AUTHOR' || $g_elem == 'NAME' || $g_elem == 'TIME'||
$g_elem == 'PUBLISHER' )
        {
            $g_books[ count( $g_books ) - 1 ][ $g_elem ] = $text;
        }
    }
```

(7) 调用上面的函数设置元素处理器、字符数据处理器，循环读取 book.xml 文件，并在页面中输出读取到的结果信息数据，代码如下：

```
    //创建解析器
    $parser = xml_parser_create();
    //元素处理器
    xml_set_element_handler( $parser, "startElement", "endElement" );
    //字符数据处理器
    xml_set_character_data_handler( $parser, "textData" );
    //打开 XML 文档
    $f = fopen( 'book.xml', 'r' );
    //循环读取 XML 文档内容
    while( $data = fread( $f, 4096 ) )
    {
        //解析读取到的 XML 文档内容
        xml_parse( $parser, $data );
    }
    //释放 XML 解析器
    xml_parser_free( $parser );
```

```
    //输出表格的开始
    echo  "<TABLE  cellSpacing=0  cellPadding=2  width='95%'  align=center
border=0>";
    //循环遍历解析后的 XML 元素名称数组
    foreach( $g_books as $book )
    {
        echo  "<TR><TD  width='400'> 书 籍 名 称 :  ".$book['NAME']."</TD><TD
width='400'>作者: ".$book['AUTHOR']."</TD></TR>";
        echo  "<TR><TD> 出 版 时 间 :  ".$book['TIME']."</TD><TD> 出 版 社 :
".$book['PUBLISHER']."</TD></TR>";
    }
    echo "</TABLE>";
```

(8) 保存 13-4.php 页面代码，并编辑静态的 HTML 代码做修饰。

(9) 新建 HTML 页面，命名为 13-4-index.htm。

(10) 13-4-index.htm 页面使用框架，将前面例子中已经编辑好的 header.htm 和 menu.htm 页面分别作为框架的顶部和左边页面，将 13-4.php 页面作为右边页面。

12.4.9 运行结果

在浏览器中运行 13-4-index.htm 页面，在页面中以表格的形式输出读取到的 XML 文件信息，如图 12-7 所示。



图 12-7　图书信息管理页面

12.4.10 实例分析



源码解析：

在上面实例中的 13-4.php 页面中，PHP 脚本首先设置 g_books 数组，它在内存中容纳所有图书信息；g_elem 变量保存正在处理的元素名称。然后定义回调函数。在这个实例中，回调函

数是 startElement()函数、endElement()函数和 textData()函数。在打开和关闭 XML 标记时，分别调用 startElement()函数和 endElement()函数；在读取开始和结束标记之间的文本内容数据时，调用了 textData()函数。

在这个实例中，开始元素名称为“BOOK”，在书籍的数组中开始一个新元素，然后，textData()函数查看当前元素是否是“AUTHOR”、“NAME”、“TIME”或者“PUBLISHER”，如果是，函数就把当前文本数据放入书籍数组中。为了让解析继续，脚本用 xml_parser_create()函数创建解析器。然后，设置回调句柄。之后，脚本读取文件把文件的大块内容发送到解析器。在文件读取之后，使用 xml_parser_free()函数释放解析器，并在末尾输出 g_books 数组的内容。

12.5　使用 DOM 处理 XML 完成学生成绩管理系统

DOM 的中文含义为文档对象模型，由一组代表 HTML 文档或者 XML 文档中不同组成部分的程序对象组成，这些对象提供了各自的属性和方法，使得应用程序开发者能够通过编写脚本程序来操纵和显示文档中相应的组件。使用 DOM 技术访问 XML 文档，具有强大的功能和灵活性，不仅可用来编写访问本地 XML 文档的程序，还可用来编写访问服务器端 XML 文档的应用程序。

在 PHP 中，可以使用 DOM 接口操作 XML 文档。在使用 DOM 接口对 XML 文档操作之前，需要将 XML 文档加载到内存中，形成一棵节点树。树上的每一个节点，就是 XML 文档的每一个标记和相应的标记内容。其实，DOM 本身是一组对象的集合，通过操纵这些对象，程序员可以操控 HTML 和 XML 数据，对文档进行读取、遍历、修改、添加和删除等操作。本节主要介绍使用内置函数库对 XML 文件进行解析。下面结合对 XML 文件的操作，介绍 DOM Functions 函数库中的函数。

 **视频教学：**光盘\videos\12\DOM 读取 XML.avi　　　　长度：6 分钟

12.5.1　基础知识——建立 XML 文件

要在 PHP 中使用 DOM 接口建立 XML 文档，首先需要创建一个 DOM 对象，也相当于在内存中创建一个树形结构，然后通过对象调用内置函数库中的相关函数建立 XML 文档。

在 PHP 中，使用 createAttribute()函数、createTextNode()函数和 createElement()函数，可以创建新的属性或新的节点并赋值；使用 appendChild()函数可以向已经创建好的节点上添加新的节点或新的属性。它们的说明如表 12-1 所示。

表 12-1　使用 DOM 创建 XML 文件所用到的函数说明

函　数	函数说明
createAttribute(name)	参数 name 表示新创建的属性的名称。该函数用于创建新的节点
createElement(name)	参数 name 的类型是字符串，这个字符串可为此元素节点规定名称。该函数用于创建元素节点，返回的是一个 Element 对象

续表

函　数	函数说明
appendChild(newchild)	参数 newchild 表示所添加的节点。该函数可向节点的子节点列表的末尾添加新的子节点，返回值为加入的节点
createTextNode(data)	参数 data 类型为字符串，规定此节点的文本。该函数用于创建文本节点，它返回的是新创建的 Text 节点值

下面创建一个实例，具体讲解使用 PHP 向 XML 文档中添加节点或属性的应用，代码如下：

```
<?php
    //XML 文档中的数据信息
    $books = array();
    $books [] = array(
        'title' => 'J2EE 开发与实战',
        'author' => '马向林',
        'publisher' => "电子信息出版社"
    );
    $books [] = array(
        'title' => 'PHP 高级应用',
        'author' => '马向林',
        'publisher' => "方立出版社"
    );
    //创建 DOM 对象
    $doc = new DOMDocument();
    $doc->formatOutput = true;
    //创建根节点
    $r = $doc->createElement( "books" );
    //新建节点 books 的属性 id
    $id=$doc->createAttribute("id");
    //创建属性值
    $idText=$doc->createTextNode("1");
    //向属性 id 赋值
    $id->appendChild($idText);
    //把新创建的 id 属性添加到 books 节点上
    $r->appendChild($id);
    //将根节点添加到 DOM 对象中
    $doc->appendChild( $r );
    //循环遍历数据数组
    foreach( $books as $book )
    {
        //创建 book 节点
        $b = $doc->createElement( "book" );
        //创建 author 节点
        $author = $doc->createElement( "author" );
        //为 author 节点创建属性值
        $author->appendChild(
            $doc->createTextNode( $book['author'] )
        );
        //将 author 节点添加到 book 节点中
```

```
        $b->appendChild( $author );
        //创建 title 节点
        $title = $doc->createElement( "title" );
        //为 title 节点创建属性值
        title->appendChild(
            $doc->createTextNode( $book['title'] )
        );
        //将 title 节点添加到 book 节点中
        $b->appendChild( $title );
        //创建 publisher 节点
        $publisher = $doc->createElement( "publisher" );
        //为 publisher 节点创建属性值
        $publisher->appendChild(
            $doc->createTextNode( $book['publisher'] )
        );
        //将 publisher 节点添加到 book 节点中
        $b->appendChild( $publisher );
        //将 book 节点添加到根节点 books 中
        $r->appendChild( $b );
    }
    //保存 XML 文件为 createBooks.xml
    $doc->save("createBooks.xml");
    echo "<a href='createBooks.xml'>查看 XML 文件</a>";
?>
```

在上面的代码中，首先创建一个 DOM 对象，然后使用 DOM 对象调用 createElement()方法创建一个名为“books”的根节点，使用$id=$doc->createAttribute("id")语句创建了属性对象 id，并使用$idText=$doc->createTextNode("1")创建了一个值为 1 的文本节点对象 idText。接下来将节点对象 idText 追加到节点对象 id 上，再将属性对象 id 追加到节点对象$r 上。

如果要创建“books”的子节点，其过程与创建“books”的属性对象 id 基本相同，只不过需要使用方法 createElement()，而不是 createAttribute()。最后，使用 appendChild()函数将创建好的节点添加到内存中的树形结构中，为根节点创建子节点和属性。运行程序，建立的 XML 文档页面效果如图 12-8 所示。

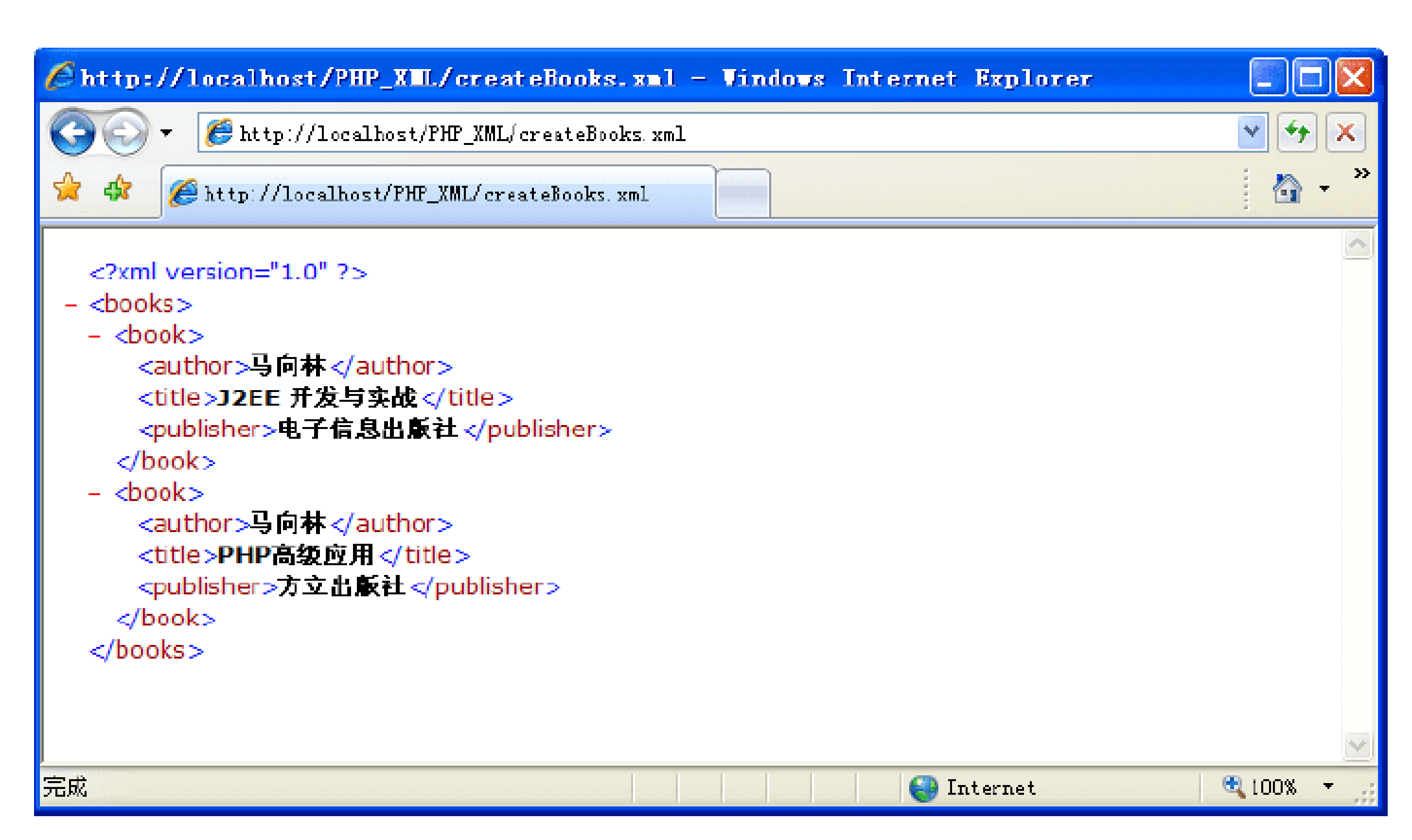


图 12-8　XML 文档页面效果图

12.5.2 基础知识——遍历节点

下面编写一个使用 DOM Functions 函数库中的函数从 XML 文件中获取数据的示例。在讲述该示例之前，首先介绍示例中需要运用到的函数库中的函数，这些函数如表 12-2 所示。

表 12-2 使用 DOM 接口遍历节点所用到的函数说明

函数或属性	函数说明
load(filename)	参数 filename 指的是要加载的文件名称，该函数用于加载一个指定 XML 文件到内存
getElementsByTagName(tagname)	该函数获取一个指定节点的节点对象，返回带有指定标签名的对象的集合
item(index)	index 参数表示索引。该函数获取指定索引值的节点对象，返回节点列表中指定索引的节点
nodeValue	该属性可根据节点的类型来设置或返回某个节点的值

通过对表 12-2 中函数的了解，现在就可以创建一个演示遍历 XML 文档的数据的示例了。首先在项目中创建 book.xml 文档，这里使用前面案例中用到的 book.xml 文档，不再创建。接下来编写 PHP 脚本代码，从而读取 XML 文档中的内容。在 PHP 程序中首先要创建一个 DOMDocument 对象，然后使用该对象加载 XML 文件，最后读取文档中的内容，代码如下所示：

```
<?php
    $doc = new DOMDocument();//创建 DOM 对象
    $doc ->load("book.xml");
    //获取 books 节点对象
    $bookstore=$doc->getElementsByTagName("book");
    //循环遍历 books 节点
    foreach ($bookstore as $book)
    {
        //获取 name 节点
        $bookname=$book->getElementsByTagName("name");
        //获取 name 节点值
        $name=$bookname->item(0)->nodeValue;
        //获取 author 节点
        $bookauthor=$book->getElementsByTagName("author");
        //获取 author 节点值
        $author=$bookauthor->item(0)->nodeValue;
        //获取 time 节点
        $booktime=$book->getElementsByTagName("time");
        //获取 time 节点值
        $time=$booktime->item(0)->nodeValue;
        //获取 publisher 节点
        $bookpublisher=$book->getElementsByTagName("publisher");
        //获取 publisher 节点值
```

```
        $publisher=$bookpublisher->item(0)->nodeValue;
        echo "书籍->$name - 作者->$author - 出版时间->$time - 出版社
->$publisher\n<br/>";
    }
?>
```

保存上面的代码，将该文件与 book.xml 文件保存在同一目录中。打开 IE 浏览器，运行页面，出现如图 12-9 所示的页面效果。

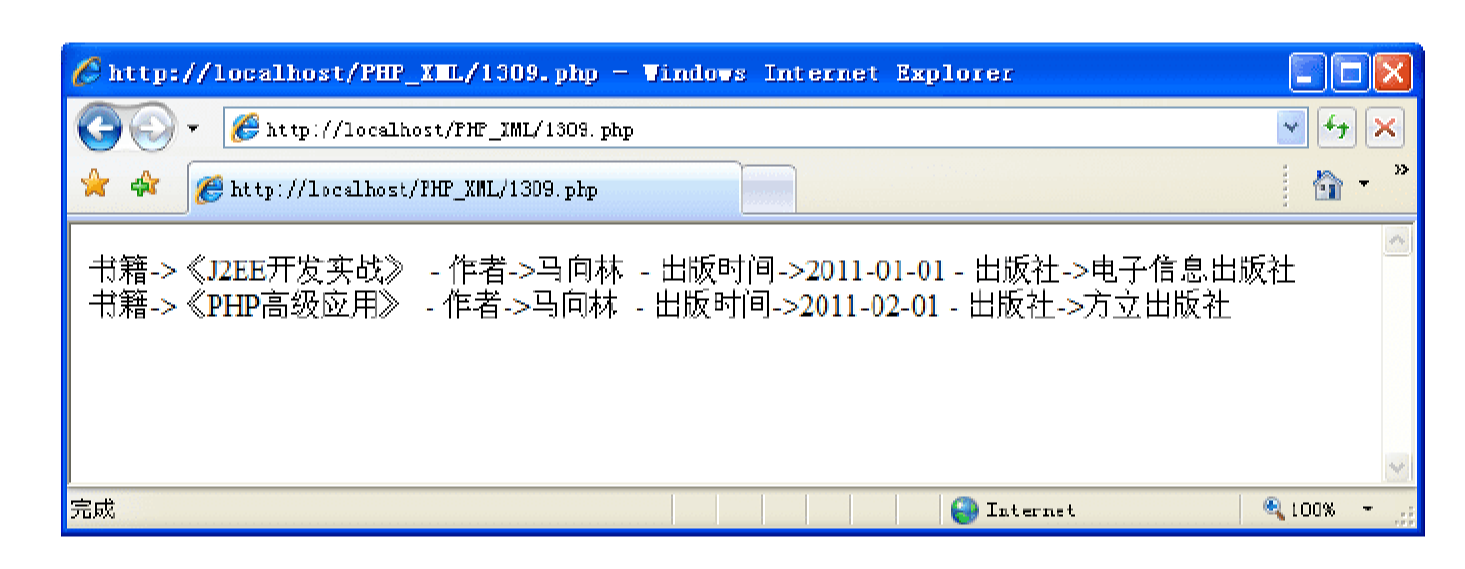


图 12-9　遍历 XML 文档显示内容

在上面的代码中，首先创建了一个 DOM 对象，然后加载一个指定的 XML 文件到内存中，并形成一个节点树，然后使用语句$bookstore=$doc->getElementsByTagName("book")创建了一个节点对象$bookstore，这时该对象就指向了节点树中节点名称为“book”的所有节点及其子节点的集合。

如果需要在页面上显示 XML 文档中的内容，也可以不遍历各个节点，直接显示到页面上，做到这点很简单，只需要使用 saveXML()函数即可，具体代码如下所示：

```
<?php
    $doc = new DOMDocument();//创建 DOM 对象
    $doc ->load("book.xml");
    print $doc->saveXML();
?>
```

12.5.3　基础知识——修改节点

使用 DOM 可以修改 XML 文档中的节点或者属性，修改这些节点其实十分简单，只需要使用一些函数就可以实现，这些函数如表 12-3 所示。

表 12-3　修改节点的函数

函数名称	函数说明
domxpath($dom)	获取一个搜索对象
query(expression)	获取指定路径的节点对象
replaceChild(new_node，old_node)	这两个参数都是必需的：new_node 指定新的节点，old_new 指定被替换的节点。该函数可将某个子节点替换为另一个，返回被替换的节点。如果替换失败，则返回 Null

续表

函数名称	函数说明
elementNode.setAttribute(name，value)	这两个参数都是必需的：name 规定要设置的属性名，value 规定要设置的属性值。该函数用于创建或改变某个新属性
save(filename)	参数 filename 指定要保存的文件名称。该函数用于将节点树保存为 filename 文件
elementNode.getAttribute(name)	参数 name 规定从中取得属性值的属性名。该函数用于获取属性 name 的值

下面创建一个实例，具体讲解如何使用这些函数修改 XML 文档中的节点和节点属性，该实例使用的 XML 文档内容如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
    <book>
        <!--作者-->
        <author>马向林</author>
        <!--书名-->
        <name sex="女">《J2EE 开发实战》</name>
        <!--出版日期-->
        <time>2011-01-01</time>
        <!--出版社-->
        <publisher>电子信息出版社</publisher>
    </book>
    <book>
        <author>马向林</author>
        <name>《PHP 高级应用》</name>
        <time>2011-02-01</time>
        <publisher>方立出版社</publisher>
    </book>
</books>
```

保存上面的 XML 代码，命名为 book1.xml。接着编辑 PHP 代码，使用表 12-3 所示的函数修改 XML 文档中的节点和节点属性，代码如下：

```
<?php
    //创建 DOM 对象
    $dom = new DOMDocument("1.0");
    //加载 XML 文件
    $dom->load("book1.xml");
    //获取搜索对象
    $xp=new DOMXPath($dom);
    //获取 author 的节点对象
    $author=$xp->query("/books/book/author")->item(0);
    //创建新的节点
    $newAuthor=$dom->createElement("author");
    //向新的节点赋值
```

```
    $newAuthor->appendChild(new DOMText("王丽丽"));
    //将$newAuthor 节点替换掉$author 节点
    $author->parentNode->replaceChild($newAuthor,$author);
    //修改属性
    $name=$xp->query("/books/book/name")->item(0);
    $name->setAttribute("sex","男");
    //将修改后的 XML 文档保存为名称为 myNewBook.xml 的文件
    $dom->save("myNewBook.xml");
    //输出修改后的 name 节点
    echo $name->getAttribute("name");
    echo "<hr/><a href='myNewBook.xml'>查看新的 XML 文档</a>";
?>
```

在上面的代码中，首先创建了一个 DOM 对象，然后将 book1.xml 文件加载到内存中，再用语句“new DOMXPath($dom)”创建了一个查询对象$xp，该对象可以获取节点树任意位置的节点。然后使用方法 query()中获取了“/books/book/author”路径下的节点对象，语句“createElement("author")”表示创建一个新的几点，节点名称为“author”。使用 appendChild()函数为新的节点添加一个文本节点，文本节点是标记之间的内容。接着使用 replaceChild()函数用新建的节点“newAuthor”替换旧的节点“author”。到此为止，就完成了修改节点内容的功能。再下面的代码使用了 setAttribute()函数将指定节点下的属性“sex”的值修改为“男”，并将修改后的文档保存为 myNewBook.xml。保存上面的代码，将页面命名为 1310.php。

打开 IE 浏览器，运行 1310.php 页面，在该页面中会显示超链接“查看新的 XML 文档”，单击该超链接，转到显示修改后的 XML 文档页面，效果如图 12-10 所示。



图 12-10　显示修改后 XML 文档内容

12.5.4　基础知识——删除节点

在 PHP 中删除 XML 文档中的指定节点或属性，只需要使用 removeChild()函数和 removeAttribute()函数。下面创建一个实例，具体说明如何使用这两个函数删除 XML 文档中的节点或属性。实例中使用的 XML 文档为前面实例中所使用的 book1.xml 文件，并稍作修改。在第二个“book”节点中添加属性 id，值为 1，即

```
<book id="1">
      <author>马向林</author>
      <name>《PHP 高级应用》</name>
      <time>2011-02-01</time>
      <publisher>方立出版社</publisher>
  </book>
```

然后创建 PHP 文件，命名为 1311.php。在页面中使用 removeChild()函数和 removeAttribute()函数删除节点和属性，代码如下：

```
<?php
    //创建 DOM 对象
    $dom=new DOMDocument("1.0");
    //加载 XML 文档
    $dom->load("book1.xml");
    //获取搜索对象
    $xp=new DOMXPath($dom);
    //搜索 id 为 1 的 book 节点
    $book=$xp->query("/books/book[@id=\"1\"]")->item(0);
    //删除节点
    $dom->documentElement->removeChild($book);
    //搜索 name 节点
    $name=$xp->query("/books/book/name")->item(0);
    //删除 name 节点的 sex 属性
    $name->removeAttribute("sex");
    $dom->save("delete.xml");
    echo "<hr/><a href='delete.xml'>查看删除节点后的 XML 文档</a>";
?>
```

在上面的代码中，首先创建一个$dom 对象，然后使用该对象调用 load()方法加载 XML 文档。接着获取要删除的节点，使用“query("/books/book[@id=\"1\"]")->item(0)”获取了 id 为 1 的 book 节点，然后使用 removeChild()函数将整个 id 为 1 的 book 节点删除。接着又获取了 name 节点，使用 removeAttribute()函数将 name 节点的 sex 属性删除。运行程序，删除节点和节点属性后的 XML 文档页面效果如图 12-11 所示。

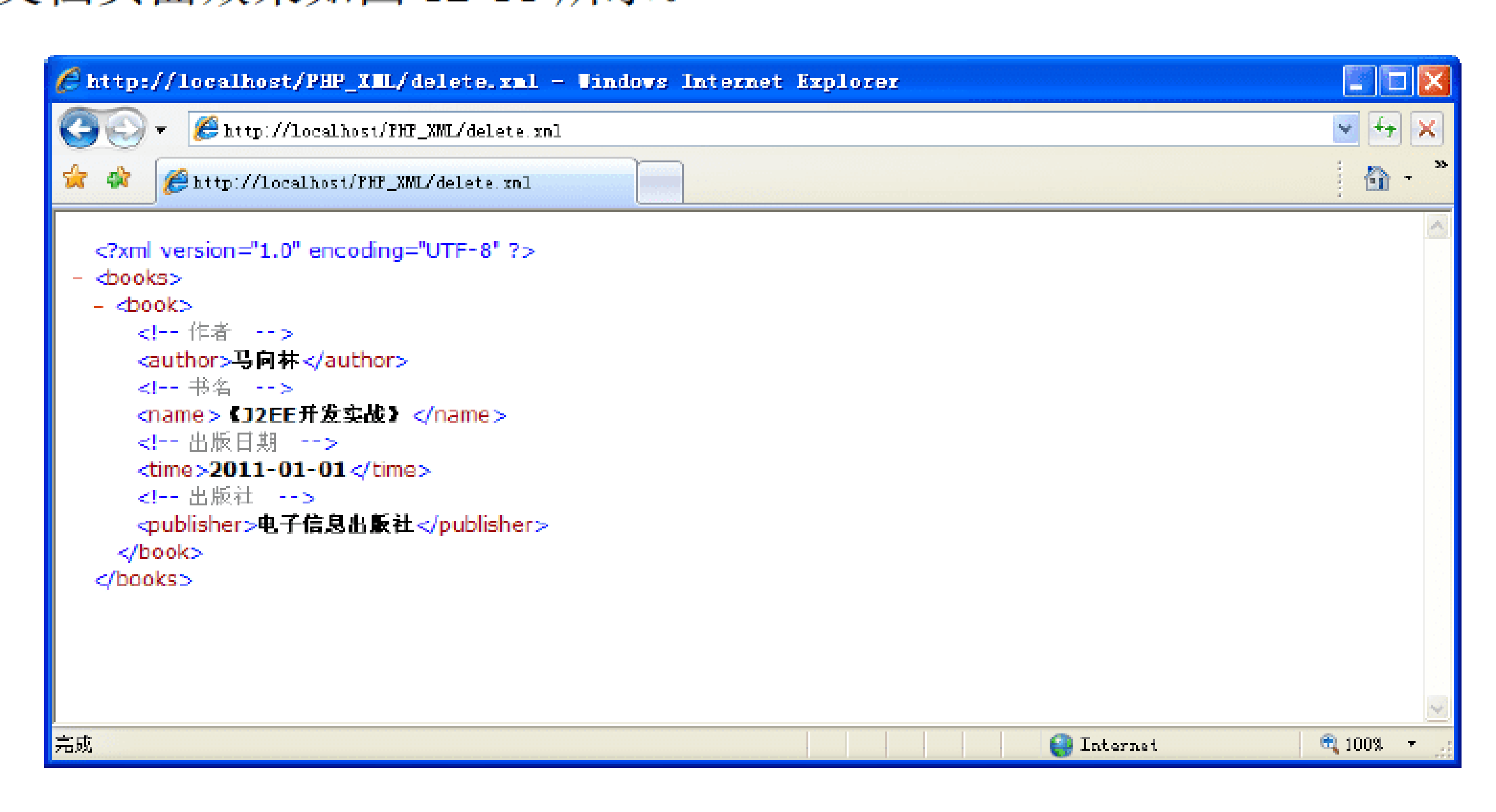


图 12-11　删除节点和节点属性后的 XML 文档页面效果

12.5.5　实例描述

在网站的后台管理系统中，功能的实现离不开添加、删除、修改、查询这 4 项操作。最近，在给客户做一个学生成绩管理系统，只要用户将学生考试信息按照一定的格式编写成 XML 文档，系统将自动解析并读取该 XML 文档数据，以表格的形式显示在页面中。当用户需要修改学生考试科目的成绩时，可以输入修改后的分数进行修改。下面就使用 DOM 操作 XML，完成学生成绩管理系统。

12.5.6　实例应用

【例 12-5】 使用 DOM 处理 XML 完成学生成绩管理系统

(1) 新建 PHP 文件，命名为 13-5.php。

(2) 在页面中遍历要读取的 XML 文档节点，读取数据并以表格的形式在页面中显示，代码如下：

```
<?php
    $doc = new DOMDocument();//创建 DOM 对象
    //判断是否是修改后加载文件
    if($_GET['update']=="update")
    {
        $doc ->load("myNewStu.xml");
    }
    else
    {
        $doc ->load("stuInfo.xml");
    }

    //获取 books 节点对象
    $stustore=$doc->getElementsByTagName("student");
    //输出表格的开始
    echo  "<TABLE  cellSpacing=0  cellPadding=2  width='95%'  align=center
border=1>";
    //输出表格标题
    echo "<TR><TD>学生姓名</TD><TD>学生性别</TD><TD>考试科目</TD><TD>分数
    </TD><TD>操作</TD></TR>";
    //循环遍历 books 节点
    foreach ($stustore as $stu)
    {
        //获取 name 节点
        $stuname=$stu->getElementsByTagName("name");
        //获取 name 节点值
        $name=$stuname->item(0)->nodeValue;
        //获取 sex 节点
        $stusex=$stu->getElementsByTagName("sex");
        //获取 sex 节点值
```

```
        $sex=$stusex->item(0)->nodeValue;
        //获取 lesson 节点
        $stulesson=$stu->getElementsByTagName("lesson");
        //获取 lesson 节点下的子节点
        foreach($stulesson as $lesson)
        {
            //获取 lessonName 节点
            $stulessonName=$lesson->getElementsByTagName("lessonName");
            //获取 lessonName 节点值
            $lessonName=$stulessonName->item(0)->nodeValue;
            //获取 lessonScore 节点
            $stulessonScore=$lesson->getElementsByTagName("lessonScore");
            //获取 lessonScore 节点值
            $lessonScore=$stulessonScore->item(0)->nodeValue;
            //输出数据
            echo "<TR><TD>".$name."</TD><TD>".$sex."</TD><TD>".$lessonName.
            "</TD><TD>".$lessonScore."</TD><TD><a href='updateScore.php?
            score=".$lessonScore."&name=".$name."&lessonname=".$lessonName."'>
            修改分数</a></TD></TR>";
        }

    }
    echo "</TABLE>";
?>
```

在上面代码的开始，判断了是否是修改后加载 XML 文件。当用户修改分数后，还是跳转至 13-5.php 页面，因此在这里需要判断。用户第一次访问时，加载 stuInfo.xml 文件，当用户修改分数后，加载修改后的 XML 文件——myNewStu.xml。

(3) 在表格的一列数据中添加了超链接，当用户单击【修改分数】超链接时，页面跳转至 updateScore.php，并将获取到的学生姓名、考试科目和科目成绩传递到 updateScore.php 页面中。接着新建 PHP 文件，并命名为 updateScore.php。

(4) 编辑 updateScore.php 页面，在页面中使用“$_GET['参数名称']”获取传递过来的参数值，即

```
<?php
    $score=$_GET['score'];
    $name=$_GET['name'];
    $lessonname=$_GET['lessonname'];
?>
```

(5) 在 updateScore.php 页面中创建供用户修改分数的表单域，将获取到的学生姓名、考试科目作为“hidden”表单项，将考试分数作为“text”的 value 输出到页面中，代码如下：

```
<form action="doUpdate.php" method="post">
    修改分数：<input type="text" name="score" value="<?php echo $score;?>"
/><br/>
    <input type="hidden" name="name" value="<?php echo $name;?>" />
    <input type="hidden" name="lessonname" value="<?php echo $lessonname;?>"
```

```
/>
    <input type="submit" value="修改分数" name="submit" />
</form>
```

(6) 当用户修改后，单击【修改分数】按钮，将表单提交至 doUpdate.php 页面中，并将学生姓名、考试科目和科目分数传递过去。接着新建 PHP 文件，命名为 doUpdate.php。

(7) 编辑 doUpdate.php 页面，获取传递过来的学生姓名、考试科目和科目分数信息，使用 DOM 解析 stuInfo.xml 文件，遍历该 XML 文档节点，根据获取到的学生姓名、考试科目信息查询相匹配的值，并修改为 updateScore.php 页面中传递过来的“score”参数的值，最后将修改的文件保存为“myNewStu.xml”，即在 13-5.php 页面中判断是否修改后加载的 XML 文档名称，然后跳转至 13-5.php 页面。代码如下：

```
<?php
    $xingming=$_POST['name'];
    $score=$_POST['score'];
    $lessname=$_POST['lessonname'];
    //创建 DOM 对象
    $dom = new DOMDocument("1.0");
    //加载 XML 文件
    $dom->load("stuInfo.xml");
    //获取搜索对象
    $xp=new DOMXPath($dom);
    //获取 student 的节点对象
    $stustore=$dom->getElementsByTagName("student");
    foreach($stustore as $stu)
    {
        //获取 name 节点
        $stuname=$stu->getElementsByTagName("name");
        //获取 name 节点值
        $studentname=$stuname->item(0)->nodeValue;
        echo $studentname;
        if($studentname==$xingming)
        {
            //获取 lesson 节点
            $stulesson=$stu->getElementsByTagName("lesson");
            foreach($stulesson as $lesson)
            {
            //获取 lessonName 节点
            $stulessonName=$lesson->getElementsByTagName("lessonName");
            //获取 lessonName 节点值
            $lessonName=$stulessonName->item(0)->nodeValue;
            if($lessonName==$lessname)
            {
                //获取 lessonScore 节点

    $studenglessonscore=$lesson->getElementsByTagName("lessonScore");
                //获取 lessonScore 节点值
                $studentscore=$studenglessonscore->item(0);
```

```
                //创建新的节点
                $newscore=$dom->createElement("lessonScore");
                //向新的节点赋值
                $newscore->appendChild(new DOMText($score));
                //将 newscore 节点替换掉$studentscore 节点

    $studentscore->parentNode->replaceChild($newscore,$studentscore);
            }

            }
        }
    }
    //将修改后的 XML 文档保存为名称为 myNewBook.xml 的文件
    $dom->save("myNewStu.xml");
    //执行完代码后，跳转至 13-5.php 页面
    header("location:13-5.php?update=update");
?>
```

(8) 新建 HTML 文件，命名为 13-5-index.htm。

(9) 在 13-5-index.htm 页面中使用 frameset 将前面案例中创建的 header.htm 和 menu.htm 页面分别作为 frameset 的顶部页面和左边菜单导航页面，将本项目中新建的 13-5.php 页面作为内容页面。

12.5.7 运行结果

在浏览器中运行 13-5-index.htm 页面，读取 stuInfo.xml 文档数据，以表格的形式显示在页面中，如图 12-12 所示。



图 12-12 读取 XML 文档数据

单击赵海波同学的《Ajax 应用开发》考试科目分数后面的【修改分数】超链接，跳转至 updateScore.php 页面，在页面中显示该考试科目的成绩“80”，如图 12-13 所示。



图 12-13　显示考试科目的成绩

当用户修改分数“80”为“99”后，单击【修改分数】按钮，表单提交至 doUpdate.php 页面，在页面中对 XML 文档中的该数据进行修改，自动跳转至 13-5.php 页面。当前 13-5.php 页面中读取的是修改后的 XML 文件——myNewStu.xml，赵海波同学的《Ajax 应用开发》考试科目分数已经修改为“99”，如图 12-14 所示。



图 12-14　修改后的 XML 文档数据

12.5.8　实例分析

源码解析：

在上面实例中的 13-5.php 页面中判断了该页面要加载的 XML 文档，当用户第一次访问该页面时，“update”的参数为 null，因此加载的是 stuInfo.xml 文件，当用户修改分数，程序执行后，该页面加载的是 myNewStu.xml 文件。然后遍历了 XML 文档中的数据，并显示在页面中。当用户单击【修改分数】超链接时，将该条数据的信息传递到了 updateScore.php 页面中，并向用户提供了修改分数的表单项。当用户单击【修改分数】按钮后，将用户输入的修改后的分数和该条数据的信息全部传递到了 doUpdate.php 页面中。

在 doUpdate.php 页面中，首先获取了要修改数据的信息；接着创建了 DOM 对象，并获取了搜索对象为 stuInfo.xml 文件；然后获取该文件中的“student”节点对象，循环遍历该节点，获取该节点下的子节点和节点值，将获取到的“name”节点值与用户要修改数据的学生姓名进行匹配，如果匹配成功，接着遍历该节点中的多个“lesson”节点，将“lesson”节点下的“lessonName”节点值与用户要修改的考试科目再次进行匹配，如果匹配成功，修改该节点下的考试成绩；最后保存修改后的 XML 文件为“myNewStu.xml”，使用 header()函数将页面自动跳转至 13-5.php 页面。

12.6　使用 SimpleXML 解析 XML 文档

SimpleXML 对象是用来临时存储 XML 数据的临时变量，对 XML 进行的操作都是通过操作 SimpleXML 对象来完成的。在 PHP 中，可以使用 PHP 提供的 SimpleXML 类库来处理非常简单的 XML 文档。本节将介绍使用 SimpleXML 处理 XML 的一些函数。

视频教学：光盘\videos\12\PHP 生成 XML.avi　　长度：5 分钟

12.6.1　基础知识——使用 SimpleXML 处理 XML

SimpleXML 是 PHP 5 的核心部分，无需安装或引用外部扩展，就可以对 XML 文档进行处理。SimpleXML 扩展为对象映射的 XML API，即 SimpleXML 把 XML 转换为对象，该对象的属性提供每个节点的元素的访问途径。可以使用数字索引或非数字索引来访问属性，还可以对从任何项中获取的文本值进行字符串转换。



SimpleXML 把 XML 文档转换为对象，元素被转换为 SimpleXMLElement 对象的单一属性。当同一级别上存在多个元素时，属性会被置于数组中，通过使用关联数组进行访问，下标对应属性名称。如果一个元素拥有多个文本节点，则按照它们被找到的顺序进行排列。

SimpleXML 类库提供了大量处理 XML 文档的函数，通过使用这些函数可以非常容易地处

理 XML 文档，下面将详细地介绍类库中的常用函数。

1．把 XML 文档载入对象中的函数

simplexml_load_file()函数将 XML 字符串载入对象中，如果失败，则返回 false。该函数的语法格式如下所示：

```
object simplexml_load_file ( string filename [, string class_name [, int
options]] )
```

由语法格式可知，该函数允许有 3 个参数，如表 12-4 所示。

表 12-4　simplexml_load_file()函数参数说明

参数名称	是否必需	参数说明
filename	必需	规定要使用的 XML 字符串
class_name	可选	规定新对象的 class
options	可选	规定附加的 Libxml 参数

该参数返回类 SimpleXMLElement 的一个对象，该对象的属性包含 XML 文档中的数据。下面使用该函数创建一个实例，代码如下：

```
<?php
    if (file_exists('bk.xml'))
    {
        $xml = simplexml_load_file('bk.xml');
        var_dump($xml);
    }
    else
    {
        exit('Error.');
    }
?>
```

保存上面的代码，在浏览器中查看运行结果，在页面中输出内容如下：

```
object(SimpleXMLElement)#1 (5)
{
    ["comment"]=> array(4)
    {
        [0]=> object(SimpleXMLElement)#2 (0) { }
        [1]=> object(SimpleXMLElement)#3 (0) { }
        [2]=> object(SimpleXMLElement)#4 (0) { }
        [3]=> object(SimpleXMLElement)#5 (0)
        { }
    }
    ["author"]=> string(9) "马向林"
    ["name"]=> string(22) "《J2EE 开发实战》"
    ["time"]=> string(10) "2011-01-01"
    ["publisher"]=> string(21) "电子信息出版社"
```

```
}
```

2. 获取指定节点的子节点函数

children()函数用于获取指定节点的子节点，语法格式如下所示：

```
class SimpleXMLElement
{
    string children(ns,is_prefix)
}
```

其中，参数 ns 和 is_prefix 都为可选的，参数 is_prefix 的值默认为 false。下面创建一个实例，具体讲解如何运用该函数。实例中运用的 XML 文档内容如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
  <book>
    <!--作者-->
      <author>马向林</author>
      <!--书名-->
      <name>《J2EE 开发实战》</name>
      <!--出版日期-->
      <time>2011-01-01</time>
      <!--出版社-->
      <publisher>电子信息出版社</publisher>
  </book>
```

下面编辑 PHP 脚本代码，使用 children()函数解析 XML 文档，代码如下所示：

```
<?php
        $xml=simplexml_load_file("bk.xml");
        foreach ($xml->children() as $book)//使用 children()函数循环遍历子节点
        {
            echo "子节点：".$book."<br/>";
        }
?>
```

保存 PHP 文件，并命名为 1313.php，在浏览器中运行该页面，在页面中输出结果如下：

```
子节点：马向林
子节点：《J2EE 开发实战》
子节点：2011-01-01
子节点：电子信息出版社
```

3. 添加子节点函数

addChild()函数向指定的 XML 节点添加节点，该函数返回一个 SimpleXMLElement 对象，这个对象表示添加到 XML 节点的子元素，该函数语法格式如下所示：

```
class SimpleXMLElement
{
    string addChild(name,value,ns)
}
```

其中，参数 name 是必需的，表示子元素的名称；参数 value 也是必需的，表示子元素的值；参数 ns 是可选的，表示子元素的命名空间。下面使用该函数创建一个示例，代码如下：

```
<?php
    $xml = simplexml_load_file("bk.xml");
    $xml->author[0]->addChild("newAuthor","王丽丽");
    foreach ($xml->author->children() as $child)
    {
      echo "Child node: " . $child;
    }
?>
```

保存并执行该段代码，结果输出：

```
Child node: 王丽丽
```

4．获取 SimpleXML 元素属性函数

attributes()函数主要用于用户获取 SimpleXML 元素属性，该函数提供了一个 XML 标签中定义的属性和值，语法格式如下所示：

```
class SimpleXMLElement
{
    string attributes(ns,is_prefix)
}
```

其中，该函数有两个参数，并且都是可选的：第一个参数表示被检索的属性的命名空间；第二个参数默认值为 false。在创建该示例之前，首先修改 XML 文档，为 author 节点添加一个 sex="男"的属性，然后编写 PHP 脚本程序，代码如下：

```
<?php
    $xml=simplexml_load_file("bk.xml");
    foreach ($xml->author[0]->attributes() as $a=>$b)
    {
        echo $a."=".$b;
    }
?>
```

保存并执行该段代码，在页面中输出：

```
sex=男
```

5．向 SimpleXML 元素添加属性函数

addAttribute()函数主要用于向 SimpleXML 元素添加一个属性。该函数无返回值，语法格式如下：

```
class SimpleXMLElement
{
    string addAttribute(name,value,ns)
}
```

该函数有 3 个参数，分别表示属性名称、属性值和属性的命名空间，其中第三个参数为可选参数。下面使用该函数创建一个实例，代码如下：

```
<?php
    $xml=simplexml_load_file("bk.xml");
    $xml->author[0]->addAttribute("age","22");     //添加属性 age=22
    foreach ($xml->author[0]->attributes() as $a=>$b)
    {
        echo $a."=".$b;
    }
?>
```

保存并执行代码，在页面中输出：

```
sex=男 age=22
```

6. 获取 XML 元素的名称函数

getName()函数从 SimpleXMLElement 对象获取 XML 元素的名称。如果成功返回当前的 XML 元素名称；如果失败，则返回 false。该函数的语法格式如下所示：

```
class SimpleXMLElement
{
    string getName()
}
```

下面创建一个实例，具体讲解如何运用该函数，代码如下所示：

```
<?php
    if(file_exists("bk.xml"))
    {
        //加载该文件
        $xml=simplexml_load_file("bk.xml");
    }
    //使用 getName()函数获取 XML 文件名称
    echo "XML 文件名称为: ".$xml->getName();
    echo "<br/>";
    foreach ($xml->children() as $book)
    {
        //获取节点名称
        echo "节点名称为: ".$book->getName();
        echo "<br/>";
    }
?>
```

保存并执行该段代码，在页面中输出内容如下：

```
XML 文件名称为: book
节点名称为: author
节点名称为: name
节点名称为: time
节点名称为: publisher
```

7．对 XML 文档的 XPath 查询函数

XPath 是一个 W3C 标准，提供了标记 XML 节点的一种基于路径的语法，相当直观。例如引用 bk.xml 文档，可以使用表达式“book/author”获取所有 author 节点。XPath 还提供了一组函数，其中，xpath()函数运行对 XML 文档的 XPath 查询。如果成功，则返回包含 SimpleXMLElement 对象的一个数组；如果失败，则返回 false。该函数的语法格式如下：

```
class SimpleXMLElement
{
    string xpath(path)
}
```

其中，参数 path 是必需的，表示 XPath 路径字符串，可以根据该参数的值选择性的获取节点值。代码如下：

```
<?php
    $xml=simplexml_load_file("bk.xml");
    $books=$xml->xpath("/book/author");
    print_r($books);
?>
```

保存并运行该页面，在页面中输出的结果如下：

```
Array ( [0] => SimpleXMLElement Object ( [0] => 马向林 ) )
```

8．创建命名空间语境函数

registerXPathNamespace()函数为下一次 XPath 查询，该函数语法格式如下所示：

```
class SimpleXMLElement
{
    string registerXPathNamespace(prefix,ns)
}
```

该函数允许有两个参数：第一个参数表示命名空间的前缀；第二个参数表示必须匹配 XML 文档中的命名空间。下面使用该函数创建一个实例，PHP 脚本程序代码如下所示：

```
<?php
    $xml=simplexml_load_file("bk.xml");
    $xml->registerXPathNamespace("msg", "http://www.itzcn.com");
    $result = $xml->xpath("author");
    foreach ($result as $message)
    {
        echo $message;
    }
?>
```

保存并执行该段代码，结果如下：

```
马向林
```

9. 获取在 XML 文档中声明的命名空间函数

getDocNamespaces()函数从 SimpleXMLElement 对象返回在 XML 文档中声明的命名空间。如果成功，返回包含命名空间名称(带有关联的 URL)的数组；如果失败，则返回 false。该函数语法格式如下：

```
class SimpleXMLElement
{
    string getDocNamespaces(recursive)
}
```

该函数有一个可选参数，表示是否返回父子节点中的所有命名空间，默认为 false。下面创建一个实例，具体讲解如何运用 getDocNamespaces()函数获取命名空间。在编写 PHP 脚本程序之前，首先需要更改 bk.xml 文档，修改后的文档内容如下所示：

```
<?xml version="1.0" encoding="UTF-8"?>
  <book xmlns:b="http://www.itzcn.com">
    <!--作者-->
      <author>马向林</author>
      <!--书名-->
      <name>《J2EE 开发实战》</name>
      <!--出版日期-->
      <time>2011-01-01</time>
      <!--出版社-->
      <publisher>电子信息出版社</publisher>
  </book>
```

接着编辑 PHP 脚本程序，使用 getDocNamespaces()函数获取节点的命名空间，代码如下：

```
<?php
    if(file_exists("bk.xml"))
    {
        $xml=simplexml_load_file("bk.xml");
    }
    print_r($xml->getDocNamespaces());
?>
```

保存上段代码，在页面中输出的结果如下所示：

```
Array ( [b] => http://www.itzcn.com )
```

SimpleXML 还支持一个同样获取命名空间的函数——getNamespace()，该函数获取在 XML 文档中使用的命名空间。如果成功，该函数返回命名空间(带有关联的 URL)的一个数组；如果失败，则返回 false。语法格式如下所示：

```
class SimpleXMLElement
{
    string getNamespace(recursive)
}
```

getNamespace()函数有一个可选参数 recursive，该参数表示是否返回父子节点中使用的所

有命名空间，默认是 false。下面使用 getNamespace()函数创建一个实例，运用的 XML 文档为修改后的 bk.xml 文档内容，该实例的 PHP 脚本程序代码如下所示：

```
<?php
    $xml=simplexml_load_file("bk.xml");
    print_r($xml->getNamespaces());
?>
```

保存并执行该段代码，在页面中输出结果如下所示：

```
Array ( )
```

10．将 DOM 节点转换为 SimpleXMLElement 对象的函数

simplexml_import_dom()函数将 DOM 节点转换为 simpleXMLElement 对象；如果失败，该函数返回 false。语法格式如下：

```
SimpleXMLElement simplexml_import_dom ( DOMNode node [, string class_name] )
```

其中，参数 node 表示要使用的 DOM 节点，为必需参数；参数 class_name 表示指定新对象的类，为可选参数。下面使用该函数创建一个实例，代码如下：

```
<?php
    $dom = new domDocument;
    $dom->loadXML('<book><author>John</author></book>');
    if (!$dom) {
        echo 'Error while parsing the document';
        exit;
    }
    $s = simplexml_import_dom($dom);
    echo $s->author;
?>
```

保存上面的代码，在浏览器中查看运行效果，在页面中输出：

```
马向林
```

12.6.2　实例描述

现在房价涨得很快，而对于那些想要买房的人来说，每天最关注的就是房产新闻，哪儿的房价跌了，哪儿的房子比较好，哪儿的房价涨了，哪个房产开盘，都是他们关注的内容。那么，下面就用 SimpleXML 解析 XML 文档来做一个房产新闻动态的页面吧！

12.6.3　实例应用

【例 12-6】 使用 SimpleXML 解析 XML 文档制作房产新闻动态页面

(1) 新建 XML 文件，命名为 rss.xml。

(2) 编辑 rss.xml 文件内容，记录房产新闻动态信息，内容如下：

```
<?xml version="1.0" encoding="utf-8"?>
<rss>
    <channel>
        <title>你想拥有一套物超所值的房子吗？</title>
        <link>http://www.itzcn.com</link>
        <description>沈阳楼市销量榜</description>
        <item>
            <title>你买或不买房价都在涨不快不慢</title>
            <link>http://school.itzcn.com/special-spid-18.html</link>
            <description>数据显示，2010 年 1 月份长春住宅的成交均价为 5270 元/m²，此后
一直十分“上进”，节节攀升。</description>
        </item>
        <item>
            <title>沈阳首套房贷利率“七五折”再现江湖</title>
            <link>http://www.itzcn.com</link>
            <description>记者另外咨询了沈阳地区的多家银行，某股份制商业银行表示，如果
是首套房贷，能够给出比 85 折更低的利率折扣。
</description>
        </item>
        <item>
            <title>房产中介面临“信任危机”租房子“难”</title>
            <link>http://www.itzcn.com</link>
            <description>您好，听说您有房子要出租，租出去了吗？……那不好意思，打扰了。”
放下电话，陈新在名单上画了一个叉，这已经是他今天画下的第 43 个叉。</description>
        </item>
    </channel>
</rss>
```

(3) 保存 rss.xml 文件。

(4) 新建 PHP 文件，命名为 13-6.php。

(5) 编辑 13-6.php 页面代码，使用 SimpleXML 解析 rss.xml 文件内容，以列表的形式显示房产新闻标题，即“title”节点中的内容，并将第一个“item”元素中的内容作为最新房产新闻显示在页面中，代码如下：

```
<?php
    $xml = simplexml_load_file("rss.xml");
    echo "<h3>".$xml->channel->title."</h3><br>";
    echo "<ul>";
    echo "<li>".$xml->channel->item[0]->title."</li>";
    echo "<li>".$xml->channel->item[1]->title."</li>";
    echo "<li>".$xml->channel->item[2]->title."</li>";
    echo "</ul>";
    echo "<hr>";
    echo "<font style='font-size:18px; color:red'>最新新闻</font><br><hr>";
    foreach ($xml->channel->item[0] as $element_name => $content) {
        print "<font color=red>".$content."</font>\n<br>";
    }
    echo "<hr>";
?>
```

(6)　保存 13-6.php 页面。

12.6.4　运行结果

在浏览器中运行 13-6.php 页面，在页面中显示房产新闻的几条标题信息，并以红色字体显示最新一条房产新闻信息，如图 12-15 所示。



图 12-15　房产新闻动态页面效果

12.6.5　实例分析



源码解析：

在上面的实例中，首先使用 simplexml_load_file()函数加载了需要解析的 XML 文档，接着使用“$xml->channel->title”读取“channel”节点下的“title”内容，然后使用 item()方法将读取到的“channel”节点下“item”节点的子节点“title”信息以无序列表的形式显示在页面中。最后遍历“channel”节点下的第一个“item”节点的子元素，并在页面中输出内容。

12.7　客户端处理 XML

前面讲解的几种方式都是在服务器端对 XML 进行处理，其实这样会占据服务器线程处理程序，给服务器造成很大的压力，减慢服务器的执行速度。有时可以在客户端处理 XML 文件信息，这些通常都采用 JavaScript 脚本程序实现。本节主要讲解如何使用脚本在客户端读取 XML

文件信息。

 **视频教学**：光盘\videos\12\客户端处理 XML.avi　　　　长度：9 分钟

12.7.1　基础知识——客户端处理 XML

JavaScript 处理 XML 主要是通过 XML DOM(XML Document Object Model)来完成的，XML DOM 定义了一套访问和操作 XML 文档的标准方法，包括遍历 XML 树以及访问、插入、删除节点的方法(函数)。在访问并处理 XML 文档之前，必须把该 XML 文件载入到 XML DOM 对象中。XML DOM 的属性如下。

- nodeName 为节点的名称。
- nodeValue 为节点的值。
- nodeType 为节点的类型。
- parentNode 为指定节点的父节点。
- childNodes 为指定节点的子节点。
- firstChild 为指定节点的第一个子节点。
- lastChild 为指定节点的最后一个子节点。
- nextSibling 为指定节点的下一个同级节点。
- previousSibling 为指定节点的上一个同级节点。
- attributes 为指定节点的属性节点。

XML DOM 的常用函数(方法)如表 12-5 所示。

表 12-5　XML DOM 的常用函数

函数名称	函数说明
createElement	创建元素节点
createAttribute	创建属性节点
createTextNode	创建文本节点
appendChild	向已经存在的节点添加子节点
removeChild	删除指定的节点
replaceChild	替换指定的节点
setAttribute	改变属性的值
getAttribute	获取属性的值
getElementsByTagName(name)	获取带有指定标签名称的所有元素

下面创建一个实例，具体讲解如何使用 JavaScript 获取节点和相应的值，该实例使用的 XML 文档为 myJs.xml 文件，XML 文档内容如下：

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
    <imags>
        <imgsrc>images/1.jpg</imgsrc>
```

```
        <title>荔枝</title>
        <url>http://www.itzcn.com</url>
    </imags>
    <imags>
        <imgsrc>images/2.jpg</imgsrc>
        <title>龙眼</title>
        <url>http://www.itzcn.com</url>
    </imags>
    <imags>
        <imgsrc>images/3.jpg</imgsrc>
        <title>柑橘</title>
        <url>http://www.itzcn.com</url>
    </imags>
    <imags>
        <imgsrc>images/5.jpg</imgsrc>
        <title>苹果</title>
        <url>http://www.itzcn.com</url>
    </imags>
</root>
```

接着新建 HTML 文件，在页面中编辑 JavaScript 脚本程序代码，如下所示：

```
<script language="javascript" type="text/javascript">
//需要读取的 xml 文件
var uRl = "myJs.xml";
var xmlDoc;
//初始化，给上述定义变量赋值
if(window.ActiveXObject)
{
    //创建 DOM 对象
    xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
    xmlDoc.async="false"
    //加载要解析的 XML 文件
    xmlDoc.load(uRl);
    //读取数据
    getvalue();
}
else if(document.implementation && document.implementation.createDocument)
{
    xmlDoc=document.implementation.createDocument("", "root", null);
    //获取 root 节点对象
    xmlDoc.load(uRl);//加载 XML 文档
}
function getvalue()
{
    //获取根节点 root 的子节点，并循环遍历
    var nodes=xmlDoc.documentElement.childNodes;
    var now=new Date();
    for(i=0;i<nodes.length;i++){
        //获取 imags 节点下的第一个子节点，是一张图片路径
```

```
        var value=nodes.item(i).childNodes.item(0).text ;
        //在页面中使用 HTML 代码显示图片
        document.write("<img src='"+value+"'/>");
        //获取第二个子节点
        document.write(nodes.item(i).childNodes.item(1).nodeName+": ")
        document.write(nodes.item(i).childNodes.item(1).text);
        document.write(nodes.item(i).childNodes.item(2).nodeName+": ")
        document.write(nodes.item(i).childNodes.item(2).text);
    }
}
</script>
```

将上述代码保存，命名为 1324.html，并且该文件要和 myJs.xml 文件保存在同一个目录下。打开浏览器，运行 1324.html 页面，出现如图 12-16 的效果。



图 12-16　客户端处理 XML

在上面的代码中，语句“new ActiveXObject("Microsoft.XMLDOM")”表示创建一个 DOM 对象，然后使用函数 load()加载指定的 XML 文件到内存中进行操作。getValue()函数中的“xmlDoc.documentElement.childNodes”表示创建根节点下的所有节点的集合。其中 childNodes 表示子节点的集合；item()方法表示依据节点具有的子节点的索引值返回该子节点；属性 nodeName 表示节点的名称，即标记的名称；属性 text 表示标记的内容。

12.7.2　实例描述

一个良好的后台管理系统是很有必要的。如果后台信息是从 XML 文档中读取的话，一个良好的 XML 文档格式也是非常重要的，下面使用 JavaScript 解析 XML 文档来实现展示商品信息功能。

12.7.3　实例应用

【例 12-7】 JavaScript 解析 XML 实现商品信息展示功能

(1) 新建 XML 文件，命名为 goodss.xml。

(2) 编辑 goodss.xml 文件内容，记录商品信息，内容如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<goodss>
    <goods>
        <name>数码相机</name>
        <type>IT 数码</type>
        <goodsunit>台</goodsunit>
        <price>3206(元)</price>
    </goods>
    <goods>
        <name>联想笔记本电脑</name>
        <type>IT 数码</type>
        <goodsunit>台</goodsunit>
        <price>3206(元)</price>
    </goods>
</goodss>
```

(3) 保存 goodss.xml 文件。新建 PHP 文件，命名为 13-7.php。

(4) 在 13-7.php 页面中编辑 JavaScript 代码，解析 goodss.xml 文件，在页面中以表格的形式显示数据。JavaScript 代码如下：

```
<script type="text/javascript">
    function createTable(xmldoc){
        //创建表格对象
        var table=document.createElement("table");
        //向表格指定属性
        table.setAttribute("width","700");
        table.setAttribute("border","1");
        table.bordercolor="#FFFFFF";
        table.cellSpacing="0";
        table.cellPadding="0";
        //设置表格边框颜色
        table.bordercolorDark="#FFFFFF";
        table.bordercolorLight="#AAAAAA";
        //指定位置创建表格
        parentId.appendChild(table);
        var header=table.createTHead();
        //设置表头背景
        header.bgColor="#EEEEEE";
        var headerrow=header.insertRow(0);
        //设置表头的高度
        headerrow.height="27";
```

```
        //向表头添加数据，指定表格的列名称
        headerrow.insertCell(0).appendChild(document.createTextNode("商品名称"));
        headerrow.insertCell(1).appendChild(document.createTextNode("类别"));
        headerrow.insertCell(2).appendChild(document.createTextNode("单位"));
        headerrow.insertCell(3).appendChild(document.createTextNode("单价"));
        //读取 XML 文件，获取“goods”节点对象
        var goodss=xmldoc.getElementsByTagName("goods");
        //循环遍历节点，读取数据
        for(var i=0;i<goodss.length;i++){
            var g=goodss[i];
            //获取商品名称数据
            var name=g.getElementsByTagName("name")[0].firstChild.data;
            //获取商品类型数据
            var type=g.getElementsByTagName("type")[0].firstChild.data;
            //获取商品单位
            var
goodsunit=g.getElementsByTagName("goodsunit")[0].firstChild.data;
            //获取商品单价
            var price=g.getElementsByTagName("price")[0].firstChild.data;
            var row=table.insertRow(i+1);
            row.height="27";
            //向表格中添加读取到的 XML 文档数据
            row.insertCell(0).appendChild(document.createTextNode(name));
            row.insertCell(1).appendChild(document.createTextNode(type));

    row.insertCell(2).appendChild(document.createTextNode(goodsunit));
            row.insertCell(3).appendChild(document.createTextNode(price));
        }

    }
    //读取 XML 函数
    function readXML(){
        //指定要解析的 XML 文档
        var url="goodss.xml";
        //如果是 IE
        if(window.ActiveXObject){
            //创建 DOM 对象
            var xmldoc=new ActiveXObject("Microsoft.XMLDOM");
            xmldoc.onreadystatechange=function(){
                if(xmldoc.readyState==4)
                    //调用创建表格函数，将读取到的数据以表格的形式展示
                    createTable(xmldoc);
                }
            //加载 XML 文档
            xmldoc.load(url);
        }
        //如果是 Mozilla
        else
if(document.implementation&&document.implementation.createDocument){
```

```
            //创建 DOM 对象
            var xmldoc=document.implementation.createDocument("","",null);
            xmldoc.onload=function(){
                //调用创建表格函数，将读取到的数据以表格的形式
                xmldoc.onload=createTable(xmldoc);
        }
        //加载 XML 文档
        xmldoc.load(url);
        }
    }
</script>
```

(5)　在页面的“body”标签中添加 onload 事件，当页面加载时，执行 readXML()函数，代码如下：

```
<BODY onLoad="readXML();">
```

(6)　在页面的<body>标签与</body>标签中添加表格，并设置 td 的 id 为上面的 JavaScript 代码中设置的表格位置容器名称“parentId”。表格代码如下：

```
<table    cellpadding="0"    cellspacing="0"    style="position:relative;
margin-top:-50px; margin-left:20px;">
    <tr>
        <td valign="top" id="parentId"> </td>
    </tr>
</table>
```

(7)　保存 13-7.php 页面，并进一步的优化。

(8)　编辑 13-7-index.htm 页面，将前面案例中的 header.htm、menu.htm 和本案例中的 13-7.php 页面使用 frameset 整合。

12.7.4　运行结果

运行 13-7-index.htm 页面，在页面中以表格的形式显示商品信息数据，如图 12-17 所示。



图 12-17　展示商品信息数据页面效果

12.7.5 实例分析


源码解析：

在上面的实例中，首先在 createTable()函数中创建了一个表格对象，并设置表头数据。接着读取“goods”节点下的子节点数据，并作为了表格单元格的数据，使用“varrow=table.insertRow(i+1)”向表格增添行，使用“row.insertCell(0).appendChild(document.createTextNode(name))”向表格中增加的行添加数据。然后定义了 readXML()函数，在这个函数中创建了 DOM 对象，并指定要解析的 XML 文件，调用了 createTable()函数，向表格中添加指定 XML 文档数据。

12.8 常见问题解答

12.8.1 PHP 如何获取 XML 节点属性值


PHP 如何获取 XML 节点属性值？
网络课堂：http://bbs.itzcn.com/thread-12479-1-1.html

这是我的 XML 文档内容：

```
<music autoPlay="true" autoNextSong="true">
    <song>
        <title>SONGA</title>
        <mp3File>asgsdgs</mp3File>
    </song>
</music>
```

请问：如何获取 music 的 autoPlay 属性值呢？
【解决办法】：假定这个 XML 是一个字符$xml，代码如下：

```
$xml = new SimpleXMLElement($xml);
$autoPlay = $xml->attributes()->autoPlay;
$autoNextSong = $xml->attributes()->autoNextSong;
```

这样就获取了 music 节点的两个属性。

12.8.2 PHP 读取 XML 列表


PHP 读取 XML 列表。
网络课堂：http://bbs.itzcn.com/thread-12480-1-1.html。

我对解析 XML 文档不太熟，假设现在有一个 XML 文档，内容如下：

```
<bcaster>
    <item item_url="http://xxx.com/1.jpg" link="http://xxx.com" itemtitle=""
/>
    <item item_url="http://xxx.com/2.jpg" link="http://xxx.com" itemtitle=""
/>
    <item item_url="http://xxx.com/3.jpg" link="http://xxx.com" itemtitle=""
/>
    <item item_url="http://xxx.com/4.jpg" link="http://xxx.com" itemtitle=""
/>
</bcaster>
```

我的问题是：怎么让 PHP 读取这 4 列数据？如果以后更新 XML 数据时，要求会同时读取更新以后的内容，而并不单单是这 4 条数据。

【解决办法】：读取 XML 文档内容的方式很多，我用 SimpleXML，目前来说它还是比较强大的，代码如下：

```
<?php
    $xml=simpleXML_load_file('xxx.xml');
    $items=$xml->item;
    foreach($items as $item)
    {
        echo "item_url:".$item['item_url'].
        "link:".$item['link']."itemtitle:".
        $item['itemtitle']."<br/>";
    }
?>
```

12.9　习　　题

一、填空题

(1) __________、属性和文本是 XML 文档中最主要的内容，用于存储和表现数据。

(2) 假设现在有两个节点对象，分别是$oldName 和$newName，如果需要将$newName 的节点替换掉$oldName 的节点，编辑 PHP 代码__________可以实现。

(3) 下面代码表示设置__________的名称为“sex”，值为“男”。

```
$name->setAttribute("sex","男");
```

二、选择题

(1) PHP 通过__________函数来设置元素处理器。

A. my_start_element_handler()

B. xml_set_element_handler()

C. xml_set_character_data_handler()

D. xml_set_processing_instruction_handler()

(2) 对下面这段代码说法正确的是：__________。

```
$doc = new DOMDocument();
$doc->formatOutput = true;
$r = $doc->createElement( "books" );
$id=$doc->createAttribute("id");
$idText=$doc->createTextNode("1");
$id->appendChild($idText);
$r->appendChild($id);
```

A. 创建了一个“books”节点，并为该节点创建了“id”子节点，值为-1。

B. 创建了一个“books”节点，值为 1。

C. 创建了一个“books”节点，并向该节点添加了一个值为 1 的“id”属性。

D. 创建了一个“books”属性，值为 1。

(3) 在 PHP 中删除 XML 文档中的指定节点或属性，只需要使用 removeChild()函数和 removeAttribute()函数，对这两个函数说法正确的是__________。

A. removeChild()函数表示删除根节点，removeAttribute()函数表示删除属性

B. removeChild()函数表示删除属性，removeAttribute()函数表示删除节点

C. removeChild()函数表示删除属性，removeAttribute()函数表示删除根节点

D. removeChild()函数表示删除节点，removeAttribute()函数表示删除属性

三、上机练习

上机练习： 制作展示图书详情信息页面。

新建 books.xml 文件，根节点为“books”，在根节点下有两个“book”子节点，并且第一个“book”子节点的 id 属性值为 1001，第二个“book”子节点的 id 属性值为 1002。在每个“book”节点下又有三个子节点，名称分别为“author”、“title”、“publisher”，并且在每个“publisher”节点中又有 id 属性，值分别为“aaa”、“bbb”。

编辑 PHP 脚本代码，解析 books.xml 文件，在页面中以列表的形式显示每个“book”节点下的数据信息，页面效果如图 12-18 所示。

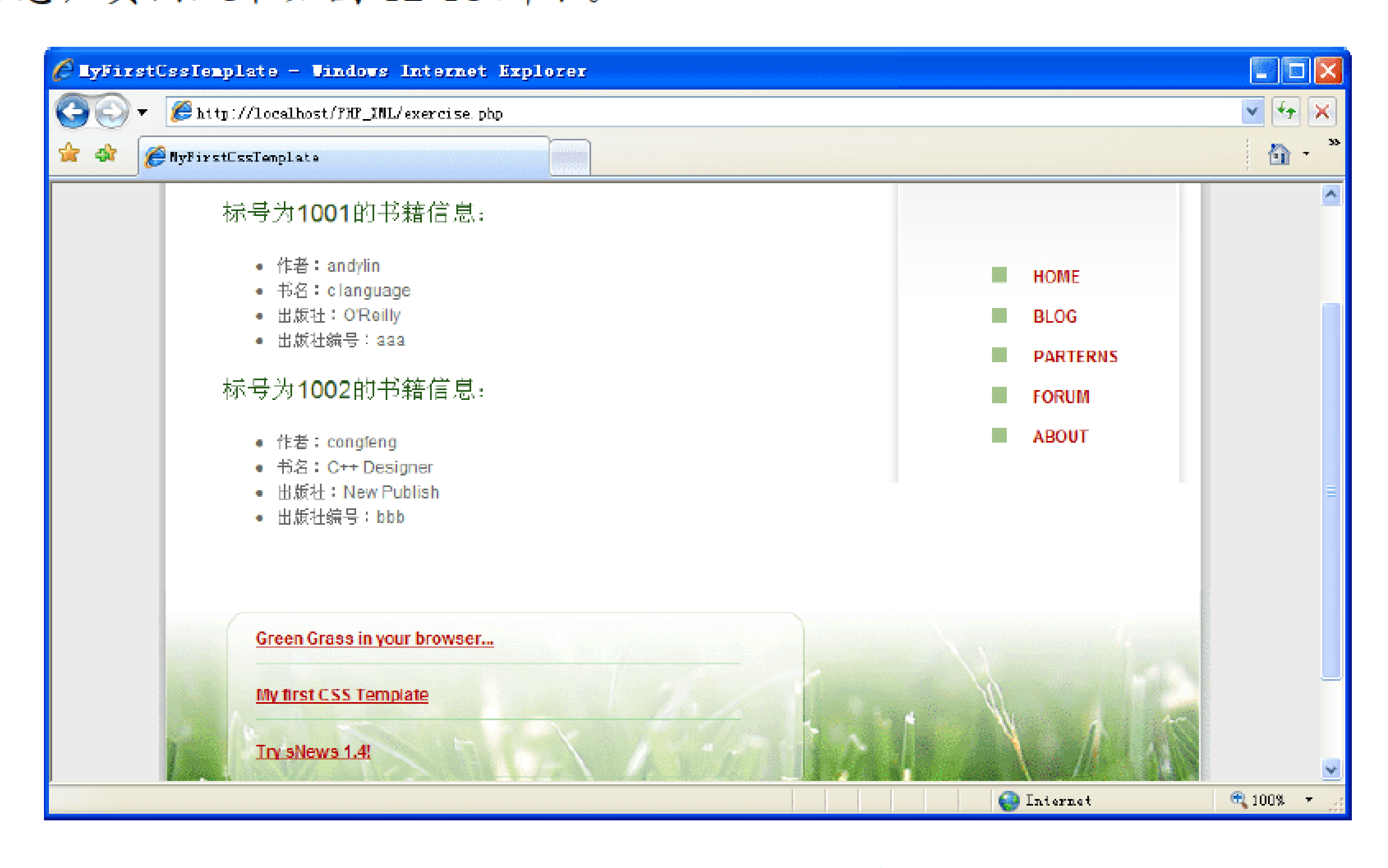
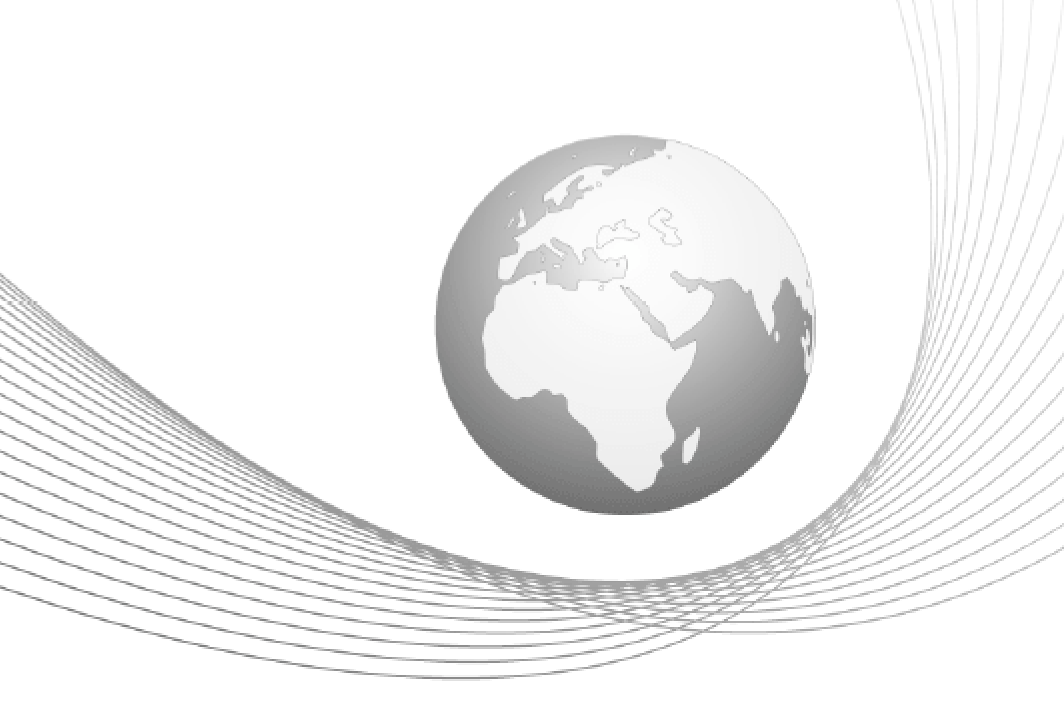


图 12-18　图书列表信息



第 13 章　PHP 与 Ajax

内容摘要：

Ajax 全称是 Asynchronous JavaScript And XML(异步 JavaScript 和 XML)，它不是一项新技术，而是很多成熟技术的集合。Ajax 是 Web 2.0 的核心，它的实现在性能上有很大提高，用户的行为经过 Ajax 引擎的处理，使得客户端可以只获取需要的数据。服务器端的组织形式将功能划分得更细，服务器只为有用的数据进行工作，提高运行效率。本章首先介绍了 Ajax 的一些基本概念，运行机制和核心内容，然后介绍了实现 Ajax 应用必不可少的核心技术——XMLHttpRequest 对象，以及如何处理服务器的响应，最后使用 Ajax 技术实现了分页显示数据功能。

学习目标：

- 理解什么是 Ajax。
- 理解 Ajax 的运行机制和核心内容。
- 理解 XMLHttpRequest 对象。
- 掌握如何使用 XMLHttpRequest 对象与服务器通信。
- 掌握 XMLHttpRequest 对象的运行周期。
- 掌握如何处理文本格式的响应。
- 掌握如何处理 XML 格式的响应。
- 掌握如何处理 JSON 格式的响应。

13.1 Ajax 实现注册表单验证

使用 Ajax 技术可以构建动态且响应更灵敏的 Web 应用程序，关键在于如何将浏览器端的 JavaScript、XML 和服务器异步通信巧妙地组合起来。

视频教学：光盘\videos\13\Ajax 概述.avi　　　　长度：4 分钟

13.1.1 基础知识——Ajax 介绍

Ajax 是异步 JavaScript 和 XML 的缩写，这只是最初的观点，由 Jesse James Garrett 创造出来。但是，Ajax 现在的覆盖面已经有了进一步的扩展，将允许浏览器与服务器通信而无需刷新当前页面的技术都包含在其中。本节将对该技术进行详细讲解。

1. 什么是 Ajax

Ajax 应用程序以异步方式连接到服务器，并取回比当前 Web 页面所能显示的内容更多的数据。这样一来，无需刷新页面就可以显示新信息。Google 是最先采用这一策略的大公司之一，它们把 Ajax 技术用在了 Gmail 和 Google Maps 里。但 Google 并不是第一个发挥出 XMLHttpRequest 对象(它是使异步连接成为可能的引擎)全部潜力的厂家。自从微软公司在 IE 5 里添加了这个对象之后，许多有经验的程序员敏锐地发现了这个对象的新用法。不久之后，Mozilla 浏览器也有了它们自己的这个对象的版本。在 XMLHttpRequest 对象出现之前，人们使用 DHTML 和 HTML 窗体和 iframe 来实现在不刷新页面的情况下改变页面的内容。

Ajax 技术主要包括：客户端脚本语言 JavaScript、异步数据获取技术 XMLHttpRequest、数据互换和操作技术 XML 和 XSLT、动态显示和交互技术 DOM 及基于标准的表示技术 XMLHTML 和 CSS 等。Ajax 极大地发掘了 Web 浏览器的潜力，开启了大量的可能性，从而有效地改善了用户操作体验。

虽然 Ajax 是一种很先进的客户端技术，但编写 JavaScript 代码(或者叫编写 Ajax 应用程序)仍令人望而生畏。即使市面上已经有了许多 IDE 可以用来编写 JavaScript 代码，程序员仍不得不面对编写客户端程序的最大挑战——浏览器的兼容问题。摆在程序员面前的事实是不同品牌的浏览器所实现的 JavaScript 总有一些细微的差异。即使是同一个浏览器，在不同的操作系统里对 JavaScript 的解释也不完全相同。因此，要想编写出一个能在所有浏览器里工作的脚本，程序员不得不在各种操作系统里使用各种浏览器去测试它，并为用来实现同一种功能的代码编写多个版本。

2. Ajax 工作原理

首先通过 XmlHttpRequest 对象向服务器发送异步请求，然后从服务器获得数据，最后用 JavaScript 来操作 DOM 而更新页面。这其中最关键的一步就是从服务器获得请求数据。要清楚这个过程和原理，我们必须对 XMLHttpRequest 有所了解。

XMLHttpRequest 是 Ajax 的核心机制，它是在 IE 5 中首先引入的，是一种支持异步请求的

技术。简单地说，就是 JavaScript 可以及时向服务器提出请求和处理响应，而不阻塞用户，达到无刷新的效果。

Ajax 技术采用的是异步交互过程，在用户与服务器之间引入一个中间的媒介，从而消除了网络交互过程中漫长的处理、等待等缺点，就是相当于在用户和服务器之间加入了一个中间层，使用户与服务器响应异步化，这样就可以把以前一些服务器负担的工作一部分转移到客户端，利用客户端闲置的处理能力来处理，减轻服务器和带宽的负担，从而达到节约 JSP 的空间及带宽租用成本的目的。

用户的浏览器在执行任务时，即加载了 Ajax 引擎，Ajax 引擎用 JavaScript 评议编写，通常存在于一个隐藏的框架中，它负责编译用户界面与服务器交互。Ajax 引擎允许用户与应用软件之间的交互过程异步进行了独立于用户与服务器之间的交互。现在，可以使用 JavaScript 调用 Ajax 引擎来代替产生一个 HTTP 的用户动作，内存中的数据编辑、页面导航、数据校验等不需要重新载入整个页面的任务就可以交给 Ajax 来执行，以此减轻服务器的负担，进而减少服务器对用户的响应时间。图 13-1 演示了 Ajax 的工作原理。

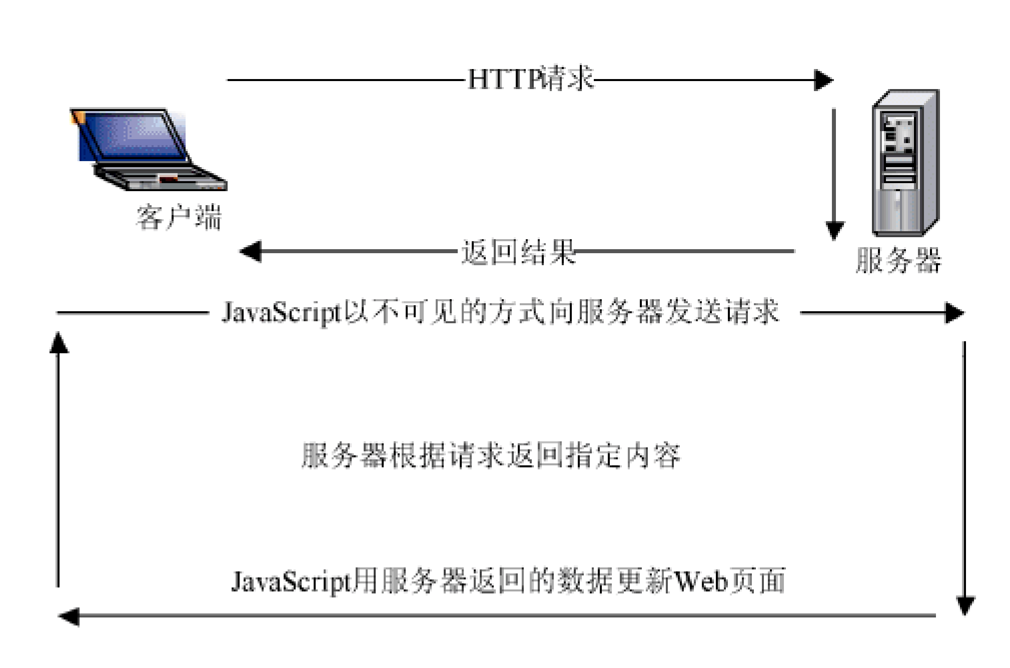


图 13-1　Ajax 的工作原理

13.1.2　实例描述

为了使编写的程序应用更加灵活健壮，程序一般会对需要用户输入信息的地方都添加了一定的数据验证功能，以避免用户输入不合法的数据，使程序意外中止。本节实例就演示了如何使用 Ajax 技术实现用户注册表单的验证功能，当用户输入数据不合法时，给出相关提示信息。

13.1.3　实例应用

【例 13-1】 Ajax 实现注册表单验证

(1) 新建一个 ajax.html 文件，在该文件中首先定义一个用户注册表单，包含用户账号、密码、确认密码三个文本输入框，一个【注册】按钮和一个【重置】按钮，用户进入注册页面后可以在注册表单中填写这些信息，具体实现代码如下所示：

```
<div id="register">
    <form>
        <table width="473">
            <tr>
```

```
                <td>账号：</td><td width="170"><input type="text"
                name="id" id="id" onChange="send_id();"/></td>
                <td width="200" rowspan="4" valign="top"><div id="sendit">
                    </div></td>
            </tr>
            <tr>
                <td>密码：</td><td><input type="password" name="password"
                id="password" onChange="send_password();"/></td>
            </tr>
            <tr>
                <td width="87">确认密码：</td>
                <td><input type="password" name="passwordagain"
                id="passwordagain" onChange="send_passwordagain();"></td>
            </tr>
            <tr>
                <td> </td>
                <td align="center">
                <input name="button" type="button" onClick="submit_form();"
                    value="注册"/>
                <input type="reset" value="重置">
                </td>
            </tr>
        </table>
    </form>
</div>
```

(2) 在 ajax.html 页面，添加如下 JavaScript 脚本，该脚本首先定义了 XMLHttpRequest()函数，通过该函数创建 XMLHttpRequest 对象，然后定义 ajax()函数在该函数中调用 XMLHttpRequest()函数得到 XMLHttpRequest 对象，判断用户提交的表单是否为 post 类型，如果是则先获取提交的参数，再调用 initReq()函数初始化并发送请求。否则直接调用 initReq()函数。在 initReq()函数中发送请求时，通过$ajax.onreadystatechange 属性指定发送请求成功时回调函数为 respHandle()。具体实现代码如下所示：

```
<script language="javascript">
    if(window.ActiveXObject)        //如果是 window 声明一个函数
    {
        function XMLHttpRequest()   //这个函数 firefox 已不用声明
        {
            try
            {
                //返回对象句柄
                return new ActiveXObject("MSXML2.XMLHTTP");
                return new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (err)
            {
                alert("XMLHttpRequest error");
            }
```

```
        }
    }
    function ajax(reqtype,url,asynch,respHandle) //声明一个 ajax 函数这个是 ajax
入口
    {
        //new 一个 xmlhttp 注意这个函数 还有前要不要带 var 表示全局变量
        $ajax=new XMLHttpRequest();
        if($ajax)                                  //检查是否 new 成功
        {
            if(reqtype.toLowerCase()!='post')   //查看是什么类型提交
            {
                initReq(reqtype,url,asynch,respHandle); //调用一个初始化函数
            }
            else
            {
                var args=arguments[4]; //如果是 post 应该还要取一个参数
                if(args!=null&&args.length>0)
                {
                    initReq(reqtype,url,asynch,respHandle,args);
                    //调用一个初始化函数
                }
            }
        }
        else
        {
            alert('ajax new error');
        }
    }

    function initReq(reqType,url,bool,respHandle)
    {
        try
        {
            //设置如果提交成功调用那个函数。注意这个是变量，是 JavaScript 创建的函数名
            $ajax.onreadystatechange=respHandle;
            $ajax.open(reqType,url,bool);       //打开地址
            if(reqType.toLowerCase()=="post")   //如果是 post 还要发送类型
            {

    $ajax.setRequestHeader("Content-Type","application/x-www-form-urlencoded;
                charset=utf-8");
                $ajax.send(arguments[4]);   //发送 post 数据
            }
            else
            {
                $ajax.send(null);            //如果是 get 就不用了
            }
        }
```

```
    catch(errv)
    {
        alert('the server at the monent');
    }
}

function handleResponse()
{
    //检查状态是否成功请求,值为 4 表示数据解析完毕,可以通过 XMLHttpRequest 对象的相
      应属性取得数据
    if($ajax.readyState==4)
    {
         //HTTP_STATUS_OK: 200: 请求处理成功
        if($ajax.status==200)
        {
            alert($ajax.responseText);// 检查返回的数据是否正确完整

document.getElementById("sendit").innerHTML=$ajax.responseText;
        }
        else
        {
            alert("$alax error");
        }
    }
}

function send_id()
{    //组装数据
    var _url="ajax.php";
    var _data="id="+document.getElementById("id").value;
    ajax("POST",_url,true,handleResponse,_data);
}
function send_password()
{
    //组装数据
    var _url="ajax.php";
    var _data="id="+document.getElementById("id").value+"&password=
    "+document.getElementById("password").value;
    ajax("POST",_url,true,handleResponse,_data);
}
function send_passwordagain()
{
    //组装数据
    var _url="ajax.php";
    var _data="id="+document.getElementById("id").value+"&password=
    "+document.getElementById("password").value+"&passwordagain="+
    document.getElementById("passwordagain").value;
    ajax("POST",_url,true,handleResponse,_data);
}
```

```
    function submit_form()
    {
        var _url="ajax.php";
        var _data="id="+document.getElementById("id").value+"&password="+
        document.getElementById("password").value+"&passwordagain="+
        document.getElementById("passwordagain").value+"&submit=1";
        ajax("POST",_url,true,handleResponse,_data);
    }
</script>
```

(3)　新建一个 ajax.php 文件，在该文件中获取表单提交传递的账号、密码和确认密码，通过判断来验证用户输入数据是否合法，具体实现代码如下所示：

```
<?PHP
    header("Content-Type:text/html;charset=utf-8");
    if(isset($_POST['id']))
    {
        $id=$_POST['id'];
        if(strlen($id)<6)
        {
            echo "<li>用户名长度小于 6</li>";
        } else if(!ereg("^[a-z][a-zA-Z0-9_]{5,16}",$id))
        {
            echo "<li>用户名不合要求</li>";
        } else {
            echo "<li>用户名合法</li>";
        }
    }

    if(isset($_POST['password']))
    {
        $pswd=$_POST['password'];
        if(strlen($pswd)<6)
        {
            echo "<li>密码长度小于 6</li>";
        } else {
            if(!ereg("^[a-zA-Z0-9_]{6,16}",$pswd))
            {
                echo "<li>密码不合要求</li>";
            } else {
                echo "<li>密码合法</li>";
            }
        }
    }

    if(isset($_POST['passwordagain']))
    {
        $pswda=$_POST['passwordagain'];
        $pswd=$_POST['password'];
```

```
        if($pswda!=$pswd)
        {
            echo "<li>两次密码不一致</li>";
        } else {
            echo "<li>密码验证成功</li>";
        }
    }

    if($_POST['submit'])
    {
        $id=$_POST['id'];
        $pswd=$_POST['password'];
        $pswda=$_POST['passwordagain'];
        echo "<li>表单验证成功</li>";
    }
?>
```

13.1.4 运行结果

实例编写完成之后，打开 IE 浏览器在地址栏中输入“http://localhost/PHP_14/ajax.html”，执行效果如图 13-2 所示。



图 13-2 注册表单验证

13.1.5 实例分析



源码解析：

本实例首先新建一个 HTML 页面 ajax.html，在页面中定义一个用户注册表单，使用户可以输入账号、密码等信息进行注册，当用户单击【注册】按钮时，将触发执行 submit_form() 函数，向服务器发送请求，服务器接受请求并进行处理，判断用户输入数据是否合法，如合法将提示用户表单验证成功，否则提示“用户输入数据不合法，请输入合法的数据。”

13.2　省市级联菜单

按需读取数据是 Ajax 技术最大的特色，这样可以有效减少数据冗余，并且减轻服务器的负担，同时还可以提高带宽的利用率。利用 Ajax 技术的这一特色，就可以实现动态加载或更新下拉列表框的问题。

视频教学：光盘\videos\13\XMLHttpRequest 对象.avi　　长度：10 分钟
光盘\videos\13\运行周期.avi　　长度：7 分钟
光盘\videos\13\初始化对象.avi　　长度：8 分钟

13.2.1　基础知识——XMLHttpRequest 对象

Ajax 应用最重要的对象是 XMLHttpRequest。XMLHttpRequest 对象承担了异步请求服务器、接收服务器响应等 Ajax 应用中最重要的工作。如果没有对 XMLHttpRequest 对象的广泛支持，就不会有现在 Ajax 技术的蓬勃发展。本节将对 XMLHttpRequest 对象进行详细讲解，包括 XMLHttpRequest 简介、XMLHttpRequest 与服务器的通信等。

1. XMLHttpRequest 简介

XMLHttpRequest 对象诞生在 IE 5.0 中，以 ActiveXObject 组件对象形式出现的。在 XMLHttpRequest 对象诞生的最初几年中，XMLHttpRequest 对象只能在 IE 中使用。直到 Mozilla 1.0 采用了 XMLHttpRequest 对象作为事实标准，XMLHttpRequest 对象使用情况才有所改观。

XMLHttpRequest 是 XMLHttp 组件的对象，通过这个对象，Ajax 可以像桌面应用程序一样只与服务器进行数据层的交换，而不必每次都刷新界面，也不必每次将数据处理的工作都交给服务器来做；这样既减轻了服务器负担又缩短了用户等待的时间。

XMLHttpRequest 可以提供不重新加载页面的情况下更新网页，页面加载后在客户端向服务器请求数据，页面加载后在服务器端接受数据，在后台向客户端发送数据。

XMLHttpRequest 得到了所有现代浏览器较好的支持。唯一的浏览器依赖性涉及 XMLHttpRequest 对象的创建。在 IE 5 和 IE 6 中，必须使用特定于 IE 的 ActiveXObject()构造函数。在 IE 中创建一个 XMLHttpRequest 对象实例，代码如下所示：

```
var xmlHttpRequestObject;
xmlHttpRequestObject = new ActiveXObject("Microsoft.XMLHTTP");
```

2. XMLHttpRequest 与服务器通信

Ajax 技术的优势体现在异步提交服务器请求，动态解析服务器响应数据这两个方面。采用 XMLHttpRequest 对象可以向服务器发送同步与异步两种方式的请求。发送同步请求的方式仅适用于开发数据量非常少的项目。如果项目数据量非常大，采用同步请求方式会使用户界面处于长时间等待服务器响应状态，这样开发出的 Web 应用可用性会大大降低。因此大多情况下采用异步请求方式与服务器进行通信。当使用 XMLHttpRequest 对象的 open()方法向服务器发送请求时，设置方法的 async 参数为 true 代表采用异步请求，为 false 代表采用同步请求方式。

使用 XMLHttpRequest 对象与服务器进行通信，大体上分为 3 个步骤，分别是创建 XMLHttpRequest 对象、发送请求和处理回调函数。具体讲解如下。

1) 创建 XMLHttpRequest 对象

下面代码中的 createXMLHttpRequest()函数演示使用 JavaScript 代码来编写跨浏览器的 XMLHttpRequest 对象实例，代码如下所示：

```
<script type="text/javascript">
    var XMLHttpRequestObject;
    function createXMLHttpRequest(){
    //在 IE 浏览器下创建 XMLHttpRequest 对象
        try {
            XMLHttpRequestObject = new ActiveXObject("MSXML2.XMLHTTP");
        } catch (exception1) {
            try {
                XMLHttpRequestObject = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (exception2) {
                XMLHttpRequestObject = false;
            }
        }
        //在 Mozilla 和 Safari 等非 IE 浏览器下创建 XMLHttpRequest 对象
        if (!XMLHttpRequestObject && window.XMLHttpRequest) {
            XMLHttpRequestObject = new XMLHttpRequest();
        }
        return XMLHttpRequestObject;
    }
</script>
```

上述代码，定义了一个全局变量 XMLHttpRequestObject 用来保存 XMLHttpRequest 对象的引用。在 createXMLHttpRequest() 函数中使用简单的分支判断逻辑确定如何创建 XMLHttpRequest 对象。if 语句中如果返回对象则为 true，返回 null 则为 false，以此得知浏览器是否支持 ActiveX 对象，也可知道浏览器是不是 Internet Explorer。如果是则使用 ActiveXObject 对象实例化一个 XMLHttpRequest 对象，并传入一个字符串指示要创建何种类型的 ActiveX 对象，否则使用 Microsoft.XMLHTTP 创建。最后，如果判断得出两种情况均不成功的话，则说明使用的不是 Internet Explorer 浏览器，将使用本地 JavaScript 对象技术创建一个 XMLHttpRequest 实例。

2) 发送请求

前面已经学习了如何创建不同浏览器中的 XMLHttpRequest 对象，要实现向服务器发送请求这一操作，需要使用 XMLHttpRequest 对象的 open()方法和 send()方法。假设有一个名为 response.xml 文件，代码如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<bestwish>
    不因换季而不想你，不因路远而不念你，不因忙碌而疏远你，更不因时间的冲刷而淡忘你。
    你永远是我心灵深处的朋友。
</bestwish>
```

XMLHttpRequest 对象的 open()方法会指定将要发出的请求。open()方法取 3 个参数：一个是指示所用方法(通常是 GET 或 POST)的字符串；一个是表示目标资源 URL 的串；一个是 Boolean 值，指示请求是否是异步的。

XMLHttpRequest 对象的 send()方法把请求发送到指定的目标资源，send()方法接受一个参数(通常是一个串或一个 DOM 对象)，这个参数作为请求体的一部分发送到目标 URL。当向 send()方法提供参数时，要确保 open()中指定的方法是 POST。如果没有数据作为请求体的一部分被发送，则使用 null。下面分别使用 GET 和 POST 方法，向服务器发送请求，代码如下所示：

```
//使用 GET 方法
xmlHttp.opean("GET","response.xml",true);
xmlHttp.send(null);
//使用 POST 方法
xmlHttp.open("POST","response.xml",true);
xmlHttp.send(null);
```

新建一个 HTML 页面，在该页面中的 body 元素中，添加一个 onload 事件，代码如下所示：

```
<body onload="sendRequest();">
</body>
```

当请求加载该 HTML 页面时，将触发 onload 事件调用执行 startRequest()函数的代码。该函数用于向服务器发送请求，代码如下所示：

```
//发送请求
function sendRequest(){
    createXMLHttpRequest();//创建 XMLHttpRequest 对象的实例
    xmlHttp.onreadystatechange = handleStateChange;
    xmlHttp.open("GET","response.xml",true);
    xmlHttp.send(null);
}
```

上述代码首先在 startRequest()函数中调用 createXMLHttpRequest()来创建 XMLHttpRequest 对象的实例，然后通过该对象实例 xmlHttp 设置 onreadystatechange 属性指定回调函数为 handleStateChange，最后调用 open()和 send()方法向服务器发送请求。

3)　处理回调函数

在发送请求函数 sendRequest()中已经设置了 XMLHttpRequest 对象的 onreadystatechange 属性指定了回调函数，在回调函数中首先需要判断确认 HTTP 就绪状态是否等于 4，即服务器响应是否已完成，然后判断 XMLHttpRequestObject.status 是否等于 200，最后执行处理服务器响应内容，实现代码如下所示：

```
function handleStateChange()
{
    if(XMLHttpRequestObject.readyState == 4){
        if(XMLHttpRequestObject.status == 200){
            alert("服务器的响应数据信息："+XMLHttpRequestObject.responseText);
        }
        else{
            alert("错误，请求页面异常！");
```

```
            }
        }
}
```

执行上述代码，在页面中可以看到一个按钮，当单击【点我会让你有意外的惊喜！】按钮时会弹出一个对话框，并在对话框中将 XML 文档中的信息显示出来，执行效果如图 13-3 所示。

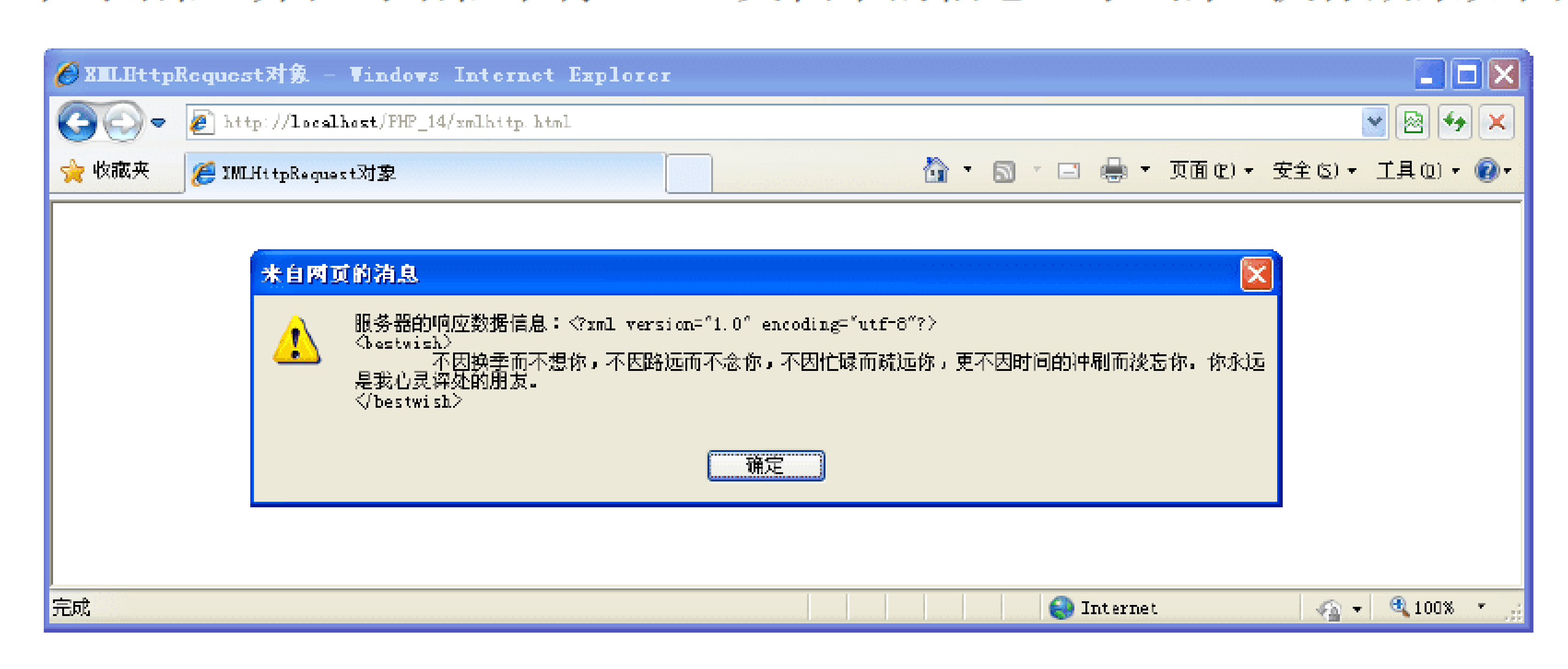


图 13-3　XMLHttpRequest 对象处理请求

13.2.2　实例描述

现在有很多网站注册时需要填写自己工作地址或家庭住址，当我们选择省份之后，在市级列表框中会自动加载罗列出我们选择的省份所包含的所有城市名称，这个功能具体是怎么实现的呢？别着急，本节实例将具体演示省市级联菜单的实现。

13.2.3　实例应用

【例 13-2】　省市级联菜单

(1)　新建一个 Category.html 页面，在该页面中定义两个下拉列表框，一个用于选择省份，一个用于根据省份加载该省份的城市。实现代码如下所示：

```
<head>
    <script language="javascript" src="Category.js"></script>
</head>
<body>
    <select id="FirstCategory1" style="width:150px;" onchange="ShowCategory
(this.value,1)">
        <option selected="selected" value="">请选择省份</option>
        <option value="10000">河南省</option>
        <option value="10001">浙江省</option>
        <option value="10002">湖南省</option>
    </select>
    <select id="SecondCategory1" style="width:200px; display:none;"></select>
</body>
```

(2)　新建一个 Category.js 文件，在该文件中首先定义一个 CreateXHR()函数来创建

XMLHttpRequest 对象，然后定义 ShowCategory()函数，当用户选择下拉列表框中的省份时发送请求，最后定义 ShowSecondCategory()函数处理客户端发送的请求，显示二级菜单，并在二级菜单中显示用户选择省份的城市。具体代码如下所示：

```
function CreateXHR()
{
    var request;
    var browser = navigator.appName;
    //使用 IE，则使用 XMLHttp 对象
    if(browser == "Microsoft Internet Explorer")
    {
        var arrVersions = ["Microsoft.XMLHttp", "MSXML2.XMLHttp.4.0",
        "MSXML2.XMLHttp.3.0", "MSXML2.XMLHttp","MSXML2.XMLHttp.5.0"];
        for (var i=0; i < arrVersions.length; i++)
        {
            try
            {
                //从中找到一个支持的版本并建立 XMLHttp 对象
                request = new ActiveXObject(arrVersions[i]);
                return request;
            }
            catch(exception)
            {
                //忽略，继续
            }
        }
    }
    else
    {
        //否则返回一个 XMLHttpRequest 对象
        request = new XMLHttpRequest();
        if(request.overrideMimeType)
        {
            request.overrideMimeType('text/xml;charset=utf-8');
        }
        return request;
    }
}
var Request = new CreateXHR(); //实例化
function ShowCategory(x,sid)   //发送请求
{
    var URL = "Category.php?value=" + x;
    varssid = sid;
    Request.open("GET",URL,true);
    Request.onreadystatechange = ShowSecondCategory;
    Request.send(null);
}
function ShowSecondCategory()  //处理请求
```

```
{
    if(Request.readyState == 4 && Request.status == 200)
    {
        document.getElementById("SecondCategory"+varssid).style.display = '';
        //读取返回的 xml 数据
        var myData = Request.responseXML.getElementsByTagName("row");
        //如果数据长度为 0 则表示没有取到数据，则不显示二级菜单
        if(myData.length == 0)
        {
            document.getElementById("SecondCategory"+varssid).style.display
                = 'none';
        }
        var myStr = new Array();    //创建一数组用来存放 text
        var myValue = new Array();  //创建一数据用来存放 value
        for(var i = 0;i<myData.length;i++)
        {
            myStr[i] = myData[i].firstChild.data;
            myValue[i] = myData[i].getAttribute("value");
        }
        document.getElementById("SecondCategory"+varssid).options.length = 0;
        for(var j = 0;j<myStr.length;j++)
        {

    document.getElementById("SecondCategory"+varssid).options[document.
            getElementById("SecondCategory"+varssid).options.length] =
            new Option(myStr[j],myValue[j]);
        }
    }
}
```

(3) 新建一个 Category.php 页面，在该页面中获取从 Category.html 页面传递过来的 value 参数，然后通过 mysql_connect()等函数连接数据库，发出 SQL 语句查询 CategoryLevel 为 second 并且 CategoryValue 为 value 的城市，最后将查询的 CategoryName 输出，具体实现代码如下所示：

```
<?php
    header("Content-Type:text/xml;charset=utf-8");
    $value = $_GET['value'];
    $link = mysql_connect("localhost","root","123456") or die("链接数据库失败
");
    mysql_select_db("xmlhttp",$link);
    mysql_query("set names utf8");
    $sql = "select CategoryID,CategoryName from category where
    CategoryLevel='second' and CategoryValue='".$value."'";
    $result = mysql_query($sql,$link);
    $data = '';
    echo $data[CategoryName];
    while(list($CategoryID,$CategoryName) = mysql_fetch_row($result))
```

```
    {
        $data .= "<row value='{$CategoryID}'>{$CategoryName}</row>";
    }
    echo "<data>".$data."</data>";
    mysql_close($link);
?>
```

13.2.4　运行结果

实例编写完成之后，启动服务器，打开 IE 浏览器在地址栏中输入“http://localhost/PHP_14/Category.html”即可看到省市菜单级联的效果，当用户选中省份，将出现二级菜单显示一级菜单中省份包含的城市，执行效果如图 13-4 所示。

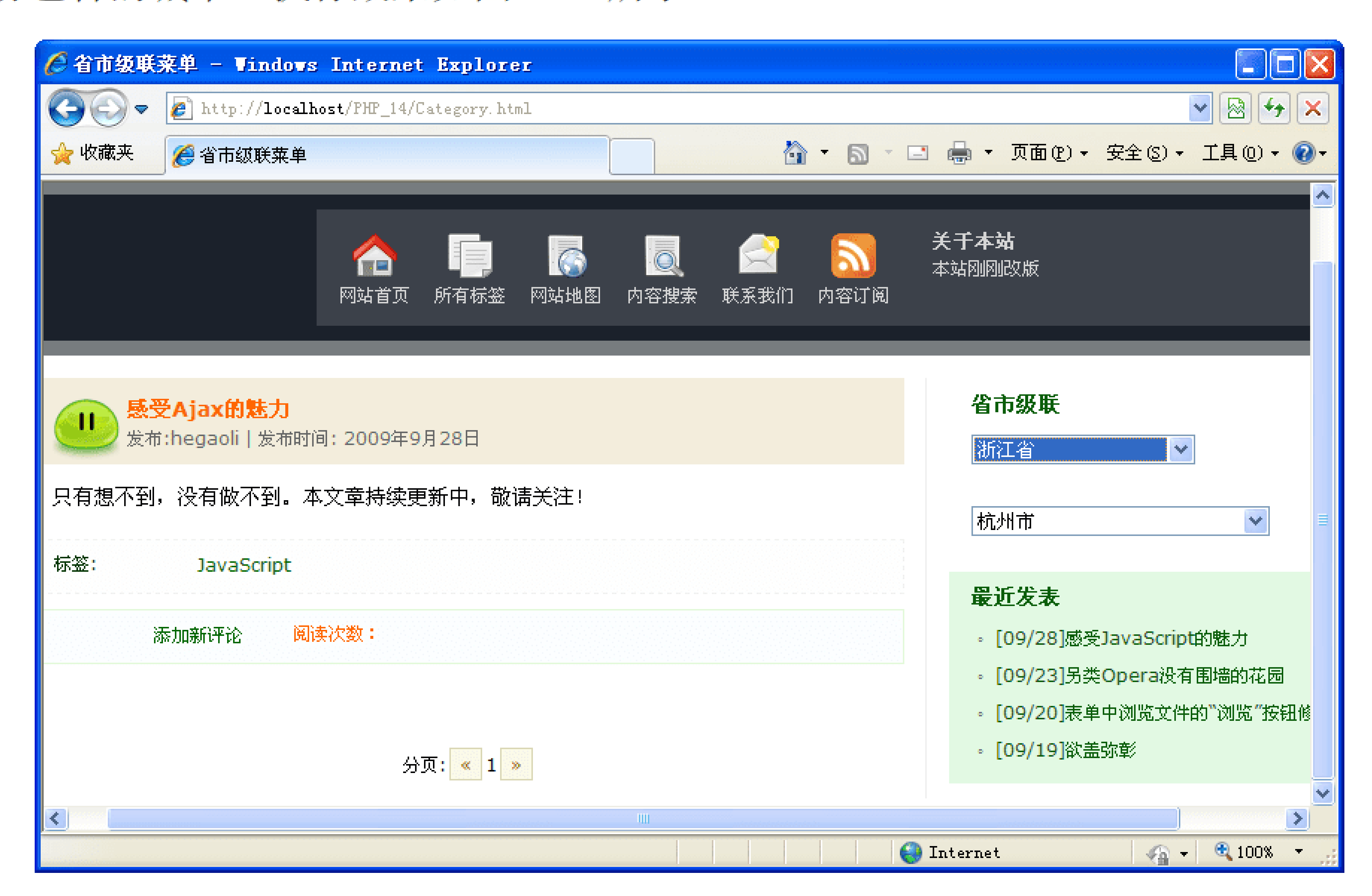


图 13-4　省市级联菜单

13.2.5　实例分析



源码解析：

本实例首先定义了一个客户端 HTML 页面，命名为 Category.html，在该页面中定义了两个下拉列表框。然后在 Category.html 页面中引入 Category.js 文件，当用户在省份列表框中选择省份时，创建 XMLHttpRequest 对象，并调用 Category.js 文件中的 ShowCategory()函数向服务器发送请求，在该函数中指定 Request.onreadystatechange = ShowSecondCategory，因此调用 ShowSecondCategory()函数处理客户请求，最后将响应获取的城市数据信息显示在二级菜单中。

13.3　获取服务器响应 JSON 格式员工信息数据

当在 Web 浏览器中输入一个 URL 时，浏览器将根据要求创建并发送请求，该请求包含所输入的 URL 以及一些与浏览器本身相关的信息。当服务器收到这个请求时将返回一个响应，该响应包括与该请求相关的信息以及位于指定 URL(如果有的话)的数据。直到浏览器解析该响应，并显示出网页(或其他资源)为止。本节将详细讲解如何使用 Ajax 技术处理服务器的各种响应。

视频教学： 光盘\videos\13\处理文本格式的响应.avi　　长度：13 分钟
光盘\videos\13\处理 XML 格式的响应.avi　　长度：6 分钟
光盘\videos\13\JSON 基本操作.avi　　长度：5 分钟

13.3.1　基础知识——处理服务器响应

在 Ajax 应用程序中需要了解五种就绪状态，分别是 0、1、2、3、4。详细介绍如下。

- 0：请求未初始化，还没有调用 open()。
- 1：请求已经建立，但是还没有发送，即还没有调用 send()。
- 2：请求已发送，正在处理中(通常现在可以从响应中获取内容头)。
- 3：请求在处理中；通常响应中已有部分数据可用了，但是服务器还没有完成响应的生成。
- 4：响应已完成；您可以获取并使用服务器的响应了。

通过 HTTP 的就绪状态可以确定请求是否已开始、是否得到了响应或者“请求/响应”模式是否已完成，以及确定读取服务器提供的响应文本或者数据是否安全。

1. 处理文本格式的响应

从客户端向服务器端发送请求后，服务器端将会根据具体的请求返回相应的数据。另外 XMLHttpRequest 对象提供了两个属性，可以用来处理服务器端返回的数据，属性具体讲解如下。

- responseText：将响应转换成字符串。
- responseXML：将响应转换成 XML 文档对象。

一般来说对于普通文本的请求，使用 responseText 属性来接受响应内容即可。如果要将接收到的响应文本内容插入到页面中，可以使用 JavaScript 调用 HTML 元素的 innerHTML 方法来实现。

下面通过一个实例具体讲解如何处理普通的文本格式响应。首先需要创建一个 XMLHttpRequest 对象，然后编写发送请求的函数，在该函数中指定处理请求的函数，具体实现代码如下所示：

```
<script type="text/javascript">
    var XMLHttpRequestObject;
```

```
    //发送请求
    function sendRequest()
    {
        var url = "text.php";
        //在 IE 下创建 XMLHttpRequest 对象
        try{
            XMLHttpRequestObject = new ActiveXObject("MSXML2.XMLHTTP");
        }catch (exception1) {
            try {
                XMLHttpRequestObject = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (exception2) {
                XMLHttpRequestObject = false;
            }
        }
        //在 Mozilla 和 Safari 等非 IE 浏览器下创建 XMLHttpRequest 对象
        if (!XMLHttpRequestObject && window.XMLHttpRequest) {
            XMLHttpRequestObject = new XMLHttpRequest();
        }
        //处理请求
        XMLHttpRequestObject.onreadystatechange = function handle (){
            if(XMLHttpRequestObject.readyState == 4){
                if(XMLHttpRequestObject.status == 200){
                    text.innerHTML = XMLHttpRequestObject.responseText;
                }
            }
        }
        XMLHttpRequestObject.open("GET",url,true);
        XMLHttpRequestObject.send(null);
    }
</script>
```

上述代码定义的 sendRequest()函数中声明一个变量，用于保存将要请求的服务器端页面的 url。该函数通过调用 createXMLHttpRequest()函数创建 XMLHttpRequest 对象，指定回调函数为 handle()，并发送请求。在 handle()函数中处理请求，首先判断 HTTP 状态 readyState 是否等于 4，即服务器响应是否完成，如果响应顺利完成则通过 responseText 属性将响应转换为字符串，并且在<div>标签中显示文本内容。

下面代码定义了一个【查看更多】按钮，通过单击该按钮向服务器发送请求，服务器处理完请求之后，将响应的文本数据发送给客户端页面显示。代码如下所示：

```
<input type="button" id="sumbit" name="button1" value="查看更多..."
onclick="sendRequest()" />
<div id="text"></div>
```

上述代码发送请求到 text.php 页面中，在该页面中首先定义 3 个变量，分别用于存储新闻的标题、发表时间和内容。然后再将其输出，最后将这些数据显示在客户端页面 id 为 text 的 div 中。实现代码如下所示：

```
<?php
```

```
    header('Content-type:text/html;charset=utf-8');
    $title="最早的图书馆——亚述巴尼拔图书馆";
    $date="2009-06-24";
    $content = "根据考古学家的发掘成果，我们已知世界上最早的图书馆也在美索不达米亚。而闻名于世的是亚述巴尼拔图书馆。这是现今已发掘的古文明遗址中，保存最完整、规模最宏大、书籍最齐全的图书馆。在时间上要比埃及著名的亚历山大图书馆早 400 年，而且由于泥版图书的特殊性，没有像亚历山大图书馆一样毁于战火，大部分都保存了下来。";
    echo("标题：".$title);
    echo("<br/>");
    echo("发表时间：".$date);
    echo("<br/>");
    echo("发表内容：".$content);
?>
```

实例代码编写完成之后，在浏览器中查看运行效果。运行客户端 HTML 页面，可以在页面中看到一个【查看更多】按钮，单击该按钮将会看到服务器响应的文本内容显示在客户端 HTML 页面中，运行效果如图 13-5 所示。



图 13-5　处理文本格式的响应数据

2. 处理 XML 格式的响应

下面通过一个实例来演示如何在服务器端创建 XML 文件，并在客户端显示 XML 文件中的数据。首先定义一个 student.html 页面，在该页面中创建 XMLHttpRequest 对象、发送请求和处理请求。具体实现代码如下所示：

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>发送 XML 请求</title>
        <script type="text/javascript">
        var xmlHttp;
        //发送请求
```

```
        function startRequest(){
            //在 IE 下创建 XMLHttpRequest 对象
            try{
                xmlHttp = new ActiveXObject("Xsxml2.XMLHTTP");
            }catch(e){
                try{
                    xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
                }catch(oc){
                    xmlHttp = null;
                }
            }
            //在 Mozilla 和 Safari 等非 IE 浏览器下创建 XMLHttpRequest 对象
            if(!xmlHttp && typeof XMLHttpRequest != "undefined"){
                xmlHttp = new XMLHttpRequest();
            }
            xmlHttp.onreadystatechange = function handleStateChange(){
            if(xmlHttp.readyState == 4){
                if(xmlHttp.status == 200){
                    var student = xmlHttp.responseXML;
                    var stu = student.getElementsByTagName("student ");
                    var str =  "<table border='1' width ='70%'><tr><td>学员号
                        </td><td>姓名
                    </td><td>年龄</td><td>年度总学分</td></tr>";
                    for(var i=0; i<stu.length;i++){

    str+="<tr><td>"+stu[i].childNodes[0].firstChild.data+"</td>";

    str+="<td>"+stu[i].childNodes[1].firstChild.data+"</td>";

    str+="<td>"+stu[i].childNodes[2].firstChild.data+"</td>";

    str+="<td>"+stu[i].childNodes[3].firstChild.data+"</td>";
                        str+="</tr>";
                    }
                    str+="</table>";
                    document.getElementById("xml").innerHTML=str;
                }
            }
        }
            xmlHttp.open("GET","student.php",true);
            xmlHttp.send(null);
        }
    </script>
    </head>
    <body onload="startRequest()">
            <h2>学员成绩信息表</h2>
            <div id="xml"></div>
    </body>
</html>
```

上述代码首先定义了一个全局变量 xmlHttp，用来存储 XMLHttpRequest 对象的引用，然后定义了一个 sendRequest() 函数，在该函数中判断根据不同的浏览器创建出相应的

XMLHttpRequest 对象，然后通过调用 XMLHttpRequest 对象的 onreadystatechange 属性来设置处理服务器响应的回调函数 handleStateChange()，在该函数中通过 xmlHttp.responseXML 属性获取服务器响应的 XML 格式数据，并创建表格使用 for 循环遍历 XML 数据中的元素，将遍历的数据元素读取到表格中，最后将表格中的数据显示在 id 为 xml 的<div>标签中。客户端设计好以后，接下来设计服务器端页面，具体实现代码如下所示：

```
<?php
    header('Content-type:text/xml');
    $xml="<?xml version='1.0' encoding='utf-8'?>";
    $data="<students>
    <student>
        <num>1001</num>
        <name>小强</name>
        <age>12</age>
        <score>89.5</score>
    </student>
    <student>
        <num>1002</num>
        <name>笑面虎</name>
        <age>13</age>
        <score>95</score>
    </student>
    <student>
        <num>1003</num>
        <name>晓航</name>
        <age>12</age>
        <score>91.5</score>
    </student>
    </students>";
    echo $xml.$data;
?>
```

上述代码在服务器端生成 XML 文件数据，执行效果如图 13-6 所示。



图 13-6　读取显示 XML 文件数据

3. 处理 JSON 格式的响应

JSON 是适用于 Ajax 应用程序的一种有效格式，原因是它使 JavaScript 对象和字符串值之

间得以快速转换。由于 Ajax 应用程序非常适合将纯文本发送给服务器端程序并对应地接收纯文本，相比不能生成文本的 API，能生成文本的 API 自然更可取，而且，JSON 让您能够处理本地 JavaScript 对象，而无需为如何表示这些对象多费心思。

XML 也可以提供文本方面的类似益处，但用于将 JavaScript 对象转换成 XML 的几个现有 API 没有 JSON API 成熟。有时，必须在创建和处理 JavaScript 对象时格外谨慎，以确保所进行的处理能与所选用的 XML 会话 API 协作。但对于 JSON，情况就大不相同，它能处理几乎所有可能的对象类型，并会返回一个非常好的 JSON 表示数据。

因此，JSON 的最大价值在于可以将 JavaScript 真的作为 JavaScript 而非数据格式语言进行处理。所有有关使用 JavaScript 对象的技巧都可以应用到代码中，而无需为如何将这些对象转变成文本而花心思。

下面通过一个实例来具体演示如何在客户端页面传递参数到服务器页面，并在服务器端生成 JSON 格式的字符串，然后将生成的 JSON 字符串再次返回到 HTML 页面。HTML 页面代码如下所示：

```
<head>
<script type="text/javascript">
    var xmlHttp;
    //发送请求
    function sendRequest(method,catid,catname,meta_title)
    {
        //创建 XMLHttpRequest 对象
        try{
            xmlHttp = new ActiveXObject("Xsxml2.XMLHTTP");
        }catch(e){
            try{
                xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
            }catch(oc){
                xmlHttp = null;
            }
        }
        //在 Mozilla 和 Safari 等非 IE 浏览器下创建 XMLHttpRequest 对象
        if(!xmlHttp && typeof XMLHttpRequest != "undefined"){
            xmlHttp = new XMLHttpRequest();
        }
        //alert(method+""+catid+""+catname+""+meta_title);
        var _url = "json.php?catid="+catid+"&catname="+catname+"&meta_title=
            "+meta_title;

        xmlHttp.onreadystatechange = function responseJSON(){
            if(xmlHttp.readyState == 4){
                if(xmlHttp.status == 200){
                    var json = eval('('+xmlHttp.responseText+')');
                    alert(json.catid+"书号指的是："+json.catname+"教程
                    主要讲解"+json.meta_title+"的具体应用开发");
                }
            }
        }
        xmlHttp.open(method,_url,true);
        xmlHttp.send(null);
    }
```

```
</script>
</head>
<body>
    <a href="#" onclick="sendRequest('get','66','php','Web Project')">PHP 教程</a>
</body>
```

上述代码定义的 sendRequest()函数有 4 个参数，用于传递向服务器页面发送数据信息。该函数首先判断根据不同的浏览器创建 XMLHttpRequest 对象，然后定义要向服务器发送请求的 URL 地址并且为其传递 catid、catname 和 meta_title 参数，并通过 XMLHttpRequest 对象的 onreadystatechange 属性设置处理响应的回调函数为 responseJSON()，在该函数中首先判断服务器响应是否完成，完成之后再判断响应是否成功，如果成功再使用 eval()函数将 xmlHttp.responseText 获取服务器响应的信息转换成 JSON 对象，然后以对话框的形式显示获取的服务器响应信息。客户端页面定义完成之后，接下来需要定义 PHP 服务器页面。实现代码如下所示：

```
<?php
    $catid = $_GET['catid'];
    $catname = $_GET['catname'];
    $meta_title = $_GET['meta_title'];
    $array = array (
        'catid' =>$catid,
        'catname' =>$catname,
        'meta_title' =>$meta_title
    );
    $request_json = json_encode($array);
    echo $request_json;
?>
```

上述代码中，定义了$catid、$catname 和$meta_title 变量来接收从客户端页面传递的参数，接着定义一个 JSON 格式的字符串$array，并且将传递过来的参数赋值给 JSON 字符串，最后输出 JSON 字符串。

这样程序就编写完成了，下面来看一下执行效果，如图 13-7 所示。



图 13-7　处理 JSON 格式的响应

13.3.2　实例描述

在很多 Web 应用程序中，需要实现获取处理服务器端数据的功能，由于 JSON 具有易于阅读、易于机器解析的特点，非常适合于作为数据传输的格式，特别是对于客户端使用 Javascript 来解析的 Web 应用。本节实例具体演示了如何获取处理服务器响应的 JSON 格式的员工信息数据。

13.3.3　实例应用

【例 13-3】 获取服务器响应 JSON 格式员工信息数据

(1) 新建一个 json.html 页面，在该页面中使用 JavaScript 脚本首先定义 createXMLHttpRequest() 函数来创建 XMLHttpRequest 对象，然后定义 startRequest() 函数，并在该函数中调用 createXMLHttpRequest()函数创建 XMLHttpRequest 对象 xmlHttp，使用该对象的 open 和 send 方法向服务器发出请求，最后定义 handleStateChange()回调函数接受处理服务器响应的 JSON 格式数据，具体代码如下所示：

```
<script type="text/javascript" language="javascript">
    var xmlHttp;
    function createXMLHttpRequest()     //创建 XMLHttpRequest 对象
    {
        try
        {
            // Firefox, Opera 8.0+, Safari
            xmlHttp=new XMLHttpRequest();
        }
        catch (e)
        {
            // Internet Explorer
            try
            {
                xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
            }
            catch (e)
            {
                xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
            }
        }
        return xmlHttp;
    }
    function startRequest(id)
    {
        createXMLHttpRequest();
        try
        {
            url="json_test.php?cid="+id;
```

```
            xmlHttp.onreadystatechange = handleStateChange;
            xmlHttp.open("GET", url, true);
            xmlHttp.send(null);
        }
        catch(exception)
        {
            alert("xmlHttp Fail");
        }
    }
    function handleStateChange()
    {
        if(xmlHttp.readyState == 4)
        {
            if (xmlHttp.status == 200 || xmlHttp.status == 0)
            {
                var result = xmlHttp.responseText;
                var json = eval("(" + result + ")");
                alert('工号:'+json.Id+'  姓名:'+json.Name+'  年龄:'+json.Age);
            }
        }
    }
</script>
```

(2) 在 json.html 页面中，定义一个【员工信息】按钮，为该按钮设置一个 onclick()事件，当用户单击该按钮，触发执行 startRequest()函数向服务器发出请求，代码如下所示：

```
<div>
    <input type="button" value="员工信息" onclick="startRequest(5)" />
</div>
```

(3) 新建一个 json.php 文件，在该文件中首先定义一个 arrayRecursive()函数存储员工信息数组$array 中所有元素指定新的 key 值，然后定义 JSON()函数将数组转换成 JSON 格式数据，并调用 json_encode()函数以供兼容中文，最后使用输出向服务器响应的数据，具体实现代码如下所示：

```
<?php
/****************************************************************
     *  使用特定 function 对数组中所有元素做处理
 * @param string &$array  要处理的字符串
 * @param string $function 要执行的函数
 * @return boolean $apply_to_keys_also  是否也应用到 key 上
 * @access public
     *************************************************************/
    function arrayRecursive(&$array, $function, $apply_to_keys_also = false)
    {
        static $recursive_counter = 0;
        if (++$recursive_counter > 1000)
        {
            die('possible deep recursion attack');
```

```
        }
        foreach ($array as $key => $value)
        {
            if (is_array($value))
            {
                arrayRecursive($array[$key], $function, $apply_to_keys_also);
            }
            else
            {
                $array[$key] = $function($value);
            }
            if ($apply_to_keys_also && is_string($key))
            {
                $new_key = $function($key);
                if ($new_key != $key)
                {
                    $array[$new_key] = $array[$key];
                    unset($array[$key]);
                }
            }
        }
        $recursive_counter--;
    }
/***************************************************************
 *  将数组转换为 JSON 字符串(兼容中文)
 * @param array $array  要转换的数组
 * @return string  转换得到的 json 字符串
 * @access public
     *************************************************************/
    function JSON($array)
    {
        arrayRecursive($array, 'urlencode', true);
        $json = json_encode($array);
        return urldecode($json);
    }
    $array = array
    (
        'Name'=>'希亚',
        'Age'=>20,
        'Id'=>$_GET['cid']
    );
    echo JSON($array);
?>
```

13.3.4　运行结果

实例编写完成之后，确保服务器能启动，打开 IE 浏览器，在地址栏中输入“http://localhost/

PHP_14/json.html”，执行效果如图 13-8 所示。

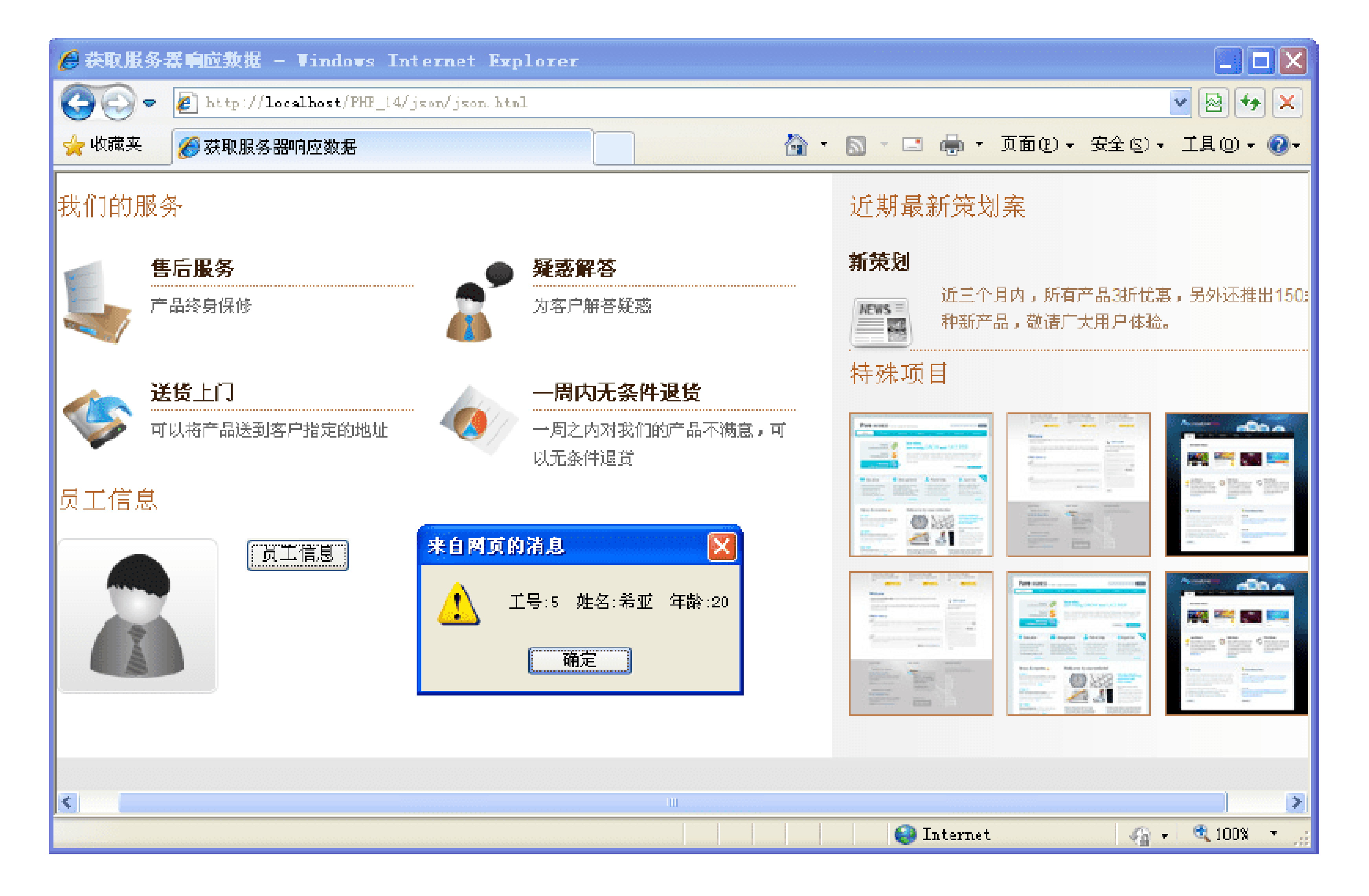
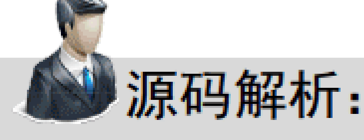


图 13-8　获取服务器响应 JSON 格式员工信息数据

13.3.5　实例分析

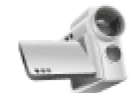


源码解析：

本节实例首先定义了一个 json.html 页面，在该页面中定义了一个【员工信息】按钮，当用户的单击该按钮将触发执行 startRequest()函数，在该函数中向服务器发送请求，并通过 XMLHttpRequest 对象 xmlHttp 的 onreadystatechange 属性指定回调函数为 handleStateChange，回调函数接收到服务器响应数据之后，在页面中弹出对话框显示获取到的 JSON 格式员工信息数据。

13.4　在 PHP 中应用 Ajax 技术实现数据分页显示

分页显示数据这一功能在各大网站中应用十分广泛，经常上网的人估计对它已经不陌生了，它使用起来很方便，那么具体是怎么实现的呢？本节将通过一个实例详细演示实现过程。

**视视频教学：**光盘\videos\13\乱码解决方案.avi　　长度：4 分钟

13.4.1　实例描述

当今这个信息化时代数据就是财富，因此尽可能合理地获取显示数据越来越重要。通过在实际项目开发中可以发现计算机的显示区域是有限的，而要显示的数据可能是无限的或不确定的。针对这些问题采用什么方法才能使数据合理完善地显示在计算机屏幕上呢？大家一定会想

到采用分页技术显示，这确实是一个很好的方法，大多数软件也都采用这种方法。采用传统的 Web 技术实现分页显示的操作都是在服务器端进行的。服务器端首先获取客户端的请求页面，然后根据请求页数获取指定的结果集，最后将结果集中的数据返回到客户端。这时返回的结果中不但包含了数据，还包含了很多 HTML 元素，就相当于客户端每一次数据更新都会重新打开一个网页，这样会造成网页打开速度较慢等情况，影响用户浏览页面的效率。本节实例采用 Ajax 技术，进行异步获取显示数据，指向服务器请求需要显示的数据进行局部内容更新，而不是整个页面。

13.4.2 实例应用

【例 13-4】 在 PHP 中应用 Ajax 技术实现数据分页显示

(1) 新建一个 index.php 页面，在该页面中首先设置每页显示条数，并连接数据库查询将要显示的数据，然后计算分页显示总页数、当前页码、上一页、下一页等数据，最后通过这些数据生成一个分页导航条，具体实现代码如下所示：

```
<?php
    header("Content-type: text/html;charset=utf-8");//header
?>
<head>
    <title>ajax page</title>
    <script language="javascript" src="ajaxpg.js"></script>
    <link rel="stylesheet" type="text/css" href="page.css">
    <style type="text/css"></style>
</head>
<body>
    <div id="result">
    <?php
        $page=isset($_GET['page'])?intval($_GET['page']):1;//获取当前页码数
        $num=3;                                     //设置每页显示 3 条数据
        $db=mysql_connect("localhost","root","123456");      //连接数据库
        mysql_select_db("xmlhttp",$db) or die("connect database error!");
                                                    //查询数据库
        mysql_query("set names utf8");              //设置字符集编码为 utf8
        $result=mysql_query("select * from brand");
        //发出查询 sql 语句，获取查询结果
        $total=mysql_num_rows($result);             //获取查询记录总条数
        $url='index.php';                           //ajax URL= index.php
                                                    //页数计算
        $pagenum=ceil($total/$num);                 //获取分页显示总页数
        $page=min($pagenum,$page);                  //当前页码
        $prepg=$page-1;                             //上一页
        $nextpg=($page==$pagenum ? 0 : $page+1);    //下一页
        $offset=($page-1)*$num;                     //得到当前页最小记录数
        $pagenav="<ul>";                            //分页导航栏
        $pagenav.="<li style=\"width:120px\9;\">
        显示第 <B>".($total?($offset+1):0)."</B>-<B>".min($offset+3,$total)."
```

```
        </B> 条记录</li><li style=\"width:90px\9;\">共 $total 条记录
             </li>";

        //pagenum = 1，即总共只显示一页
        if($pagenum < 1)
             return false;
        $pagenav.="<li> <a href=javascript:dopage('result','$url?page=1');>
            首页</a></li> ";
        if($prepg)
            $pagenav.="<li><a href=javascript:dopage('result','$url?page=$prepg');>
            前页</a></li> "; else $pagenav.=" <li>前页</li> ";
        for($i = 1;$i<= min($page+10,$pagenum);$i++){
            if($page  == $i){
                $pagenav .= "<li>".$i."</li> ";
            }else{
                $pagenav .= "<li>
                <a href=javascript:dopage('result','$url?page=".$i."');>
                    ".$i."</a></li> ";
            }
        }
        if($nextpg)
            $pagenav.="<li><a href=javascript:dopage('result','$url?page=
                $nextpg');>
            后页</a> </li>"; else $pagenav.=" <li>后页</li> ";
        if($page == $pagenum)
            $pagenav.="<li>尾页</a></li> ";else $pagenav.="<li>
            <a href=javascript:dopage('result','$url?page=$pagenum');>尾页
            </a></li> ";
        $pagenav.="<li style=\"width:60px\9;\">第 $page 页</li>
        <li style=\"width:60px\9;\">共 $pagenum 页</li></ul>";
        //page error
        if($page>$pagenum){
            echo "Error : Can Not Found The page ".$page;
            exit;
        }
    ?>
    </div>
    <div id="results">
    <?php
        $info=mysql_query("select * from brand limit ".($page-1)*$num.", $num",$db)
        or die("没有数据");   //获取相应页数所需要显示的数据
        while($it=mysql_fetch_array($info)){
            echo $it['sp_brand'];
            echo "<br>";
        }
        echo"<br>";
        echo $pagenav;//输出分页导航
    ?>
    </div>
```

```
</body>
```

(2) 新建一个 ajaxpg.js 脚本文件，在该脚本文件中声明一个存储 XMLHttpRequest 对象引用的变量 http_request，定义一个 send_request()函数，在该函数中判断根据不同浏览器初始化 XMLHttpRequest 对象，并使用 http_request.open("GET",url,true)，设置发送请求方式、URL，及是否同步执行，具体实现代码如下所示：

```
// JavaScript 脚本
var http_request=false;
function send_request(url)      //初始化，指定处理函数，发送请求的函数
    http_request=false;
    //开始初始化 XMLHttpRequest 对象
    if(window.XMLHttpRequest)       //Mozilla 浏览器
    {
        http_request=new XMLHttpRequest();
        if(http_request.overrideMimeType) //设置 MIME 类别
        {
            http_request.overrideMimeType("text/xml");
        }
    }
    else if(window.ActiveXObject)   //IE 浏览器
        {
        try{
            http_request=new ActiveXObject("Msxml2.XMLHttp");
        }catch(e){
            try{
                http_request=new ActiveXobject("Microsoft.XMLHttp");
            }catch(e){}
        }
    }
    if(!http_request) //异常，创建对象实例失败
    {
        window.alert("XMLHttp fail! ");
        return false;
    }
    http_request.onreadystatechange=processrequest;
    //确定发送请求方式，URL，及是否同步执行下段代码
    http_request.open("GET",url,true);
    http_request.send(null);
}
//处理返回信息的函数
function processrequest(){
    if(http_request.readyState==4) //判断对象状态
        {
        if(http_request.status==200) //信息已成功返回，开始处理信息
        {
            document.getElementById("results").style.display="none";
            document.getElementById(reobj).innerHTML=http_request.responseText;
        }
```

```
            else//页面不正常
            {
                alert("您所请求的页面不正常！");
            }
        }
}
function dopage(obj,url){
    document.getElementById(obj).innerHTML="<font color='green' font-size='12'>
    正在读取数据...</font>";
    send_request(url);
    reobj=obj;
}
```

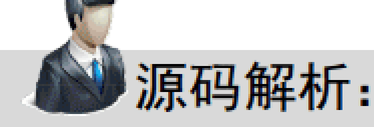
13.4.3　运行结果

实例编写完成之后，确定服务器已启动，打开 IE 浏览器在地址栏中输入“http://localhost/PHP_14/fenye/index.php”，执行效果如图 13-9 所示。



图 13-9　应用 Ajax 技术实现分页

13.4.4　实例分析


源码解析：

本实例首先创建了一个 index.php 页面，在该页面设置分页显示每页将要显示的记录条数，并定义一个分页显示导航条，引入了 ajaxpg.js 文件，该文件创建了 XMLHttpRequest 对象，通过该对象向服务器发送请求，当服务器处理完请求之后，再将响应结果返回给客户端 index.php 页面，在该页面中分页显示服务器响应的数据。

13.5　常见问题解答

13.5.1　xmlhttp.open 中 true 与 false 的区别



xmlhttp.open 中 true 与 false 的区别。
网络课堂：http://bbs.itzcn.com/thread-13111-1-1.html。

刚开始接触 Ajax 技术，使用了 xmlhttp.open("GET",URL,true)函数，但不明白 xmlhttp.open 中的 true 与 false 的区别。请解答。

【解决办法】：true 表示采用异步传输。false 表示采用同步传输。

采用异步传输，页面不等待 Ajax 访问请求返回就继续显示，如果采用同步传输，页面要等到 Ajax 访问请求返回后才会继续，这样显示就会慢些，一般建议采用异步传输，即设为 true。

13.5.2　PHP 与 Ajax 验证表单问题



PHP 与 Ajax 验证表单问题。
网络课堂：http://bbs.itzcn.com/thread-13112-1-1.html。

当用户输入注册用户名时，利用 Ajax 在动态页面检查该用户名是否存在，一般情况下，这种用户注册验证分为两步，第一步利用 JS 检查用户名是否为空，若满足 JS 的判断要求；第二步在提交数据的时候，检查是不是符合用户名，是否存在这个要求(利用 Ajax+PHP 的返回值判断)。代码如下：

```
<form id="myform" action="do.php" method="POST">
    用户名：<input type="text" id="username" onchange="checkname()/>
    //checkname()的代码在下面，主要是利用 AJAX+PHP 检测用户是否已经存在
    <input type="submit" onclick="return checkform()" value="提交" />
    //checkform()主要是最后提交的时候检测整个表单是否符合要求，
    //后再决定页面是否跳转到 do.php 中去
</form>
function checkname()
{
    var http_Ajax=createAjax();
    var username=document.getElementById('username').value;
    var url="../register/checkName.php? username="+username;
    //检测用户是否存在的页面，如果不存在返回 1
    url=encodeURI(url); //编码。确保中文可以输入
    url=encodeURI(url); //编码
    http_Ajax.open("GET",url,true);
    http_Ajax.send(null);
    http_Ajax.onreadystatechange=function()
    {
        if(http_Ajax.readyState==4)
```

```
        {
            var text=http_Ajax.responseText;
            if(text==1)     //可以注册
            {
                return true;
            }
            else            //不可以注册
            {
                return false;
            }
        }
    }
}
function checkform()
{
    if(document.myform.username.value == "")
    {
        alert("用户名输入有误");
        document.myform.username.focus();
        return false;
        if(!checkname())        //checkname()返回 true 表示可以注册
        {
            return false;
        }
        return true;            //最后如果都满足。则执行 do.php
    }
}
```

请大家帮我看看问题出在哪儿，即使用户名已经存在了，也执行了 do.php 并不是停留在表单的页面。

【解决办法】：建议你在 JavaScript 里面使用一个“全局”变量，比如 isUsernameAvailable，并设置它的初始值为 false，而且在提交表单时检查它的值，然后才能提交的。并且只有 Ajax 检查用户名不存在才能赋值为 true。

另外还需注意 http_Ajax.onreadystatechange=function(){}应该放到 http_Ajax.open("GET",url, true);的前面。

13.5.3 Ajax 实现不刷新的一段 JS，Firefox 正常而 IE 报错



Ajax 实现不刷新的一段 JS，Firefox 正常 IE 报错。
网络课堂：http://bbs.itzcn.com/thread-13113-1-1.html。

这段代码在 Firefox 中运行正常，但在 IE 中会提示 document.getElementById ("kind").innerHTML=XMLHttp.responseText;有错误，不知道是怎么回事，求高手解决，代码如下所示：

```
function send(){
```

```
    if (XMLHttp.readyState==4){
        if (XMLHttp.status==200){
            document.getElementById("kind").innerHTML=XMLHttp.responseText;
        }else{
            alert("error");
        }
    }
}
```

【解决办法】：你需要先判断浏览器是否为 IE 系列。不是的话，你可以用你的这些代码；是的话，你需要用 ActiveX 对象来创建 XMLHttpRequest 实例。下面代码能帮你获取适合浏览器的 XmlHttpRequest 实例。代码如下所示：

```
function getXMLHttpRequest()
{
    var xmlHttpRequest;
    if (window.XMLHttpRequest)
    {
        xmlHttpRequest = new XMLHttpRequest();
    }
    else
    {
        if (window.ActiveXObject)
        {
            xmlHttpRequest = new ActiveXObject("Microsoft.XMLHTTP");
        }
    }
    return xmlHttpRequest;
}
```

13.6　习　　题

一、填空题

(1)　Ajax 的全称是____________，它不是一项新技术，而是很多成熟技术的集合。

(2)　____________是 Ajax 的核心机制，简单地说，就是 JavaScript 可以及时向服务器提出请求和处理响应，而不阻塞用户，达到无刷新的效果。

(3)　XMLHttpRequest 对象的运行过程：XMLHttpRequest 对象初始化→发送请求→____________→服务器返回→客户端接受→修改客户端页面内容。

二、选择题

(1)　Ajax 术语是由____________公司或组织最先提出的？

A．Google　　B．IBM

C．Adaptive Path　　D．Dojo Foundation

(2)　以下____________Web 应用不属于 Ajax 应用？

A. Hotmail　　B. GMaps

C. Flickr　　D. Windows Live

(3) 以下__________技术不是 Ajax 技术体系的组成部分？

A. XMLHttpRequest　　B. DHTML

C. CSS　　D. DOM

(4) XMLHttpRequest 对象有__________个返回状态值？

A. 3　　B. 4　　C. 5　　D. 6

(5) 创建一个对象 obj，该对象包含一个名为"name"的属性，其值为"value"。以下__________JavaScript 代码无法得到上述的结果？

A.
```
    var obj = new Object();
obj["name"] = "value";
```

B.
```
    var obj = new Object();
obj.prototype.name = "value";
```

C.
```
    var obj = {name : "value"};
```

D.
```
    var obj = new function()
{
    this.name = "value";
}
```

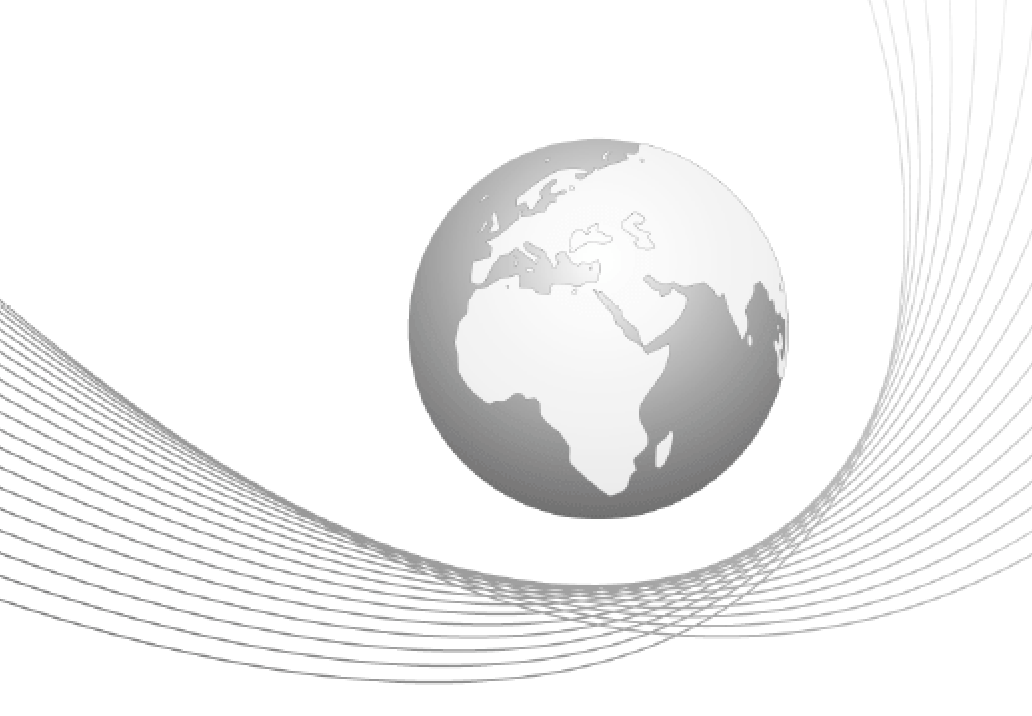
三、上机练习

上机练习： 使用 Ajax 技术获取手机号码。

练习要求：新建一个 HTML 页面，在该页面中首先创建一个 XMLHttpRequest 对象，然后指定需要请求的 url 地址，使用 XMLHttpRequest 对象发送请求，并通过 XMLHttpRequest 对象的 onreadystatechange 属性指定回调函数处理服务器响应的电话号码，最后新建一个 PHP 页面，在该页面中输出要向客户端 HTML 页面响应的电话号码数据。执行效果如图 13-10 所示。



图 13-10　获取电话号码



第 14 章　企业宣传网

内容摘要：

21 世纪，是个竞争的时代，企业如果想在激烈的竞争中脱颖而出，做好对外的宣传是至关重要的。企业宣传作为一个沟通企业与社会、企业与消费者的桥梁，在现代商战中的重要作用很明显，甚至与企业的生存和发展密切相关，有效的宣传已经成为企业促进的生产销售并提高竞争力的有效途径。因此就出现了“企业宣传网”的热门话题，本章主要通过使用 PHP 技术来完成一个企业宣传网站的建设和开发。从开发中学习更多有关 PHP 的使用技术和方法。

学习目标：

- 掌握 PHP 变量的使用。
- 字符串与 HTML 相互转换。
- 掌握 PHP 自动生成表单。
- 熟悉 PHP 与 JavaScript 集成。
- 熟练读取与写入文件。
- 掌握 PHP 处理文件上传和下载。
- 掌握传输会话 ID 及获得会话 ID。
- 掌握如何使用会话存储数据。
- 熟练使用 PHP 连接 MySQL 数据库。

14.1 系 统 分 析

系统分析包括系统需求分析和系统可行性分析。从系统需求分析，我们主要了解系统的性能需求和功能需求；从系统可行性分析中我们主要了解系统的经济效益和系统构建时技术的可行性。

企业宣传网对于一家公司或者企业来说是至关重要的一部分，企业宣传网站能够给企业的发展和销量带来很大的变化。但是到现在还是有很多企业使用平面形式来宣传公司的产品。随着信息化社会的发展，企业会进入国际化轨道，因此会有很多企业使用网络宣传自己的产品和业务。使用企业宣传网，可以大大地减少企业宣传的成本，也会提高宣传效益。

14.1.1 设计分析

需求分析活动其实本来就是个与客户交流，正确引导客户将自己的实际需求用较为适当的技术语言进行表达(或者由相关技术人员帮助表达)，以明确项目目的的过程。这个过程中也包含了对要建立的网站基本功能和模块的确立和策划活动。

所以在设计网站之前我们应注意：网站设计时的可增长性功能需求、安全性及可靠性的要求、网站的日常维护、网站的运行环境、网站页面总体风格以及美工效果、主页面和次级页面数量、内容管理及录入任务的分配和各种页面特殊效果等因素。

经过分析大概可以设计出该系统的主要模块，再经过对系统整体的分析可以设计具体模块的应用，如图 14-1 为企业宣传网管理员用例图，图 14-2 企业宣传网会员用例图。

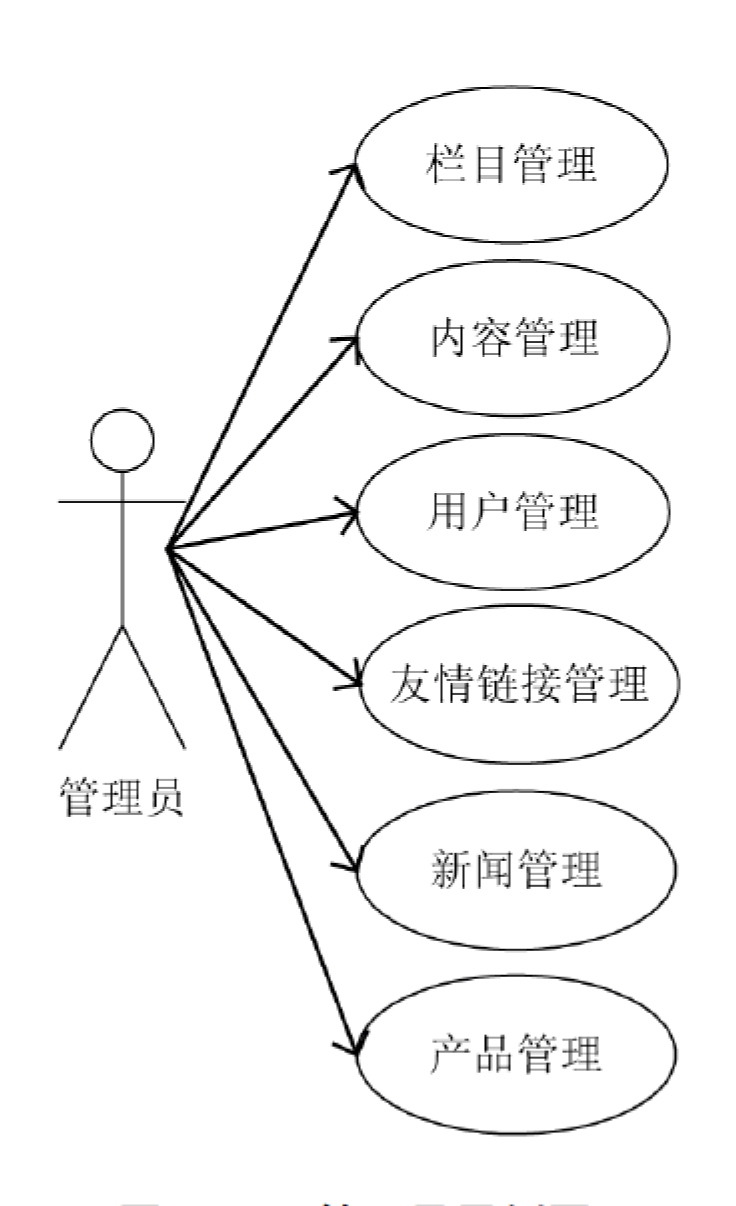


图 14-1　管理员用例图

修改个人信息

咨询问题

添加收藏

修改密码

会员

图 14-2　会员用例图

企业宣传网功能模块如图 14-3 所示。

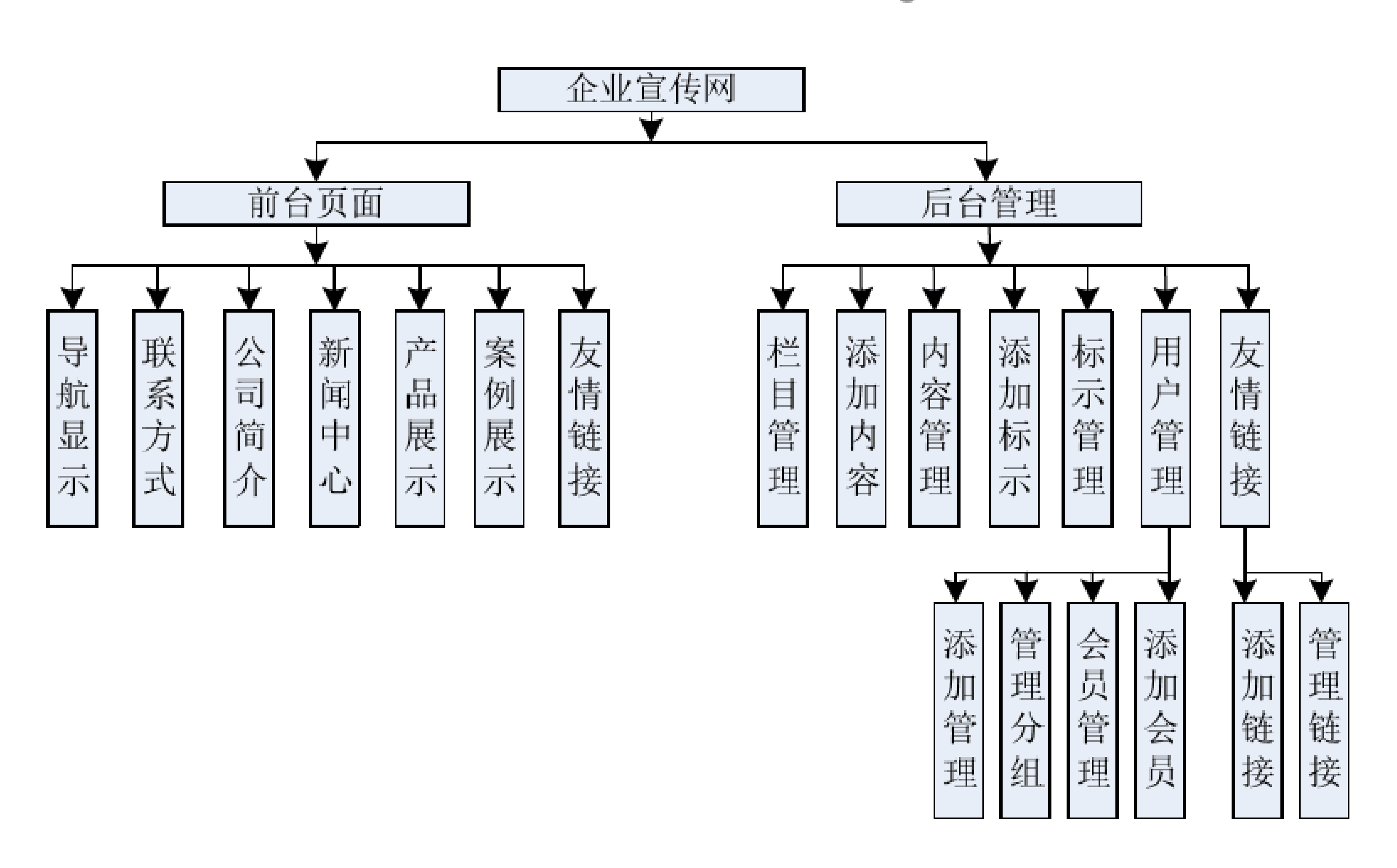


图 14-3　功能模块图

14.1.2　功能分析

通过以上的设计分析，可以把企业宣传网站划分为前台模块和后台模块，前台模块用来显示信息内容，有导航模块、联系方式模块、公司简介模块、新闻中心模块、产品展示模块、案例展示模块、友情链接模块；后台有网站设置模块、内容管理模块、模型模块、用户管理模块、友情链接模块、表单管理模块、咨询管理模块等，会员管理模块修改个人信息、咨询问题、添加收藏、修改密码。接下来将对常用的几个模块功能进行分析。

1．首页模块

- 导航显示：导航的功能是方便用户对一些网站或者工具的应用，因此使用导航来代替转向的页面，在该实例中，同样使用导航功能，该导航的内容是从数据库能读取出来显示在首页面中，用户可以通过后台对该导航进行管理。
- 联系方式：在企业宣传网中，多数都会有联系方式这个模块，因为当客户想要您的产品时，却很难找到联系你们的方式，因此为了提高营销效应，将在首页面中显示联系方式模块，该模块内容通过后台进行管理。
- 公司简介：公司简介模块主要的作用就是对该企业的一些信息简单地介绍，这样方便客户对该公司更加了解和信任。
- 新闻中心：在有些情况下，可能公司里面的产品需要进行调整，降价、涨价等情况，有时不能及时通知到用户，因此通过站内新闻的形式告知客户一些站内的信息内容，可以减少和客户之间的误会，在后台可以通过编写新闻内容将新闻信息显示在前台新闻中心列表中。
- 案例展示：案例展示是企业的产品已经和某些企业或者用户已经交易成功，并且对方已经在使用你的产品，它的使用主要是让更多的客户了解和在展示的实例中操作产品的实用性和效应等功能。
- 产品展示：它的作用同样是用来展示产品信息的功能，但是它有一个特殊的使命，就是将该公司的所有产品全部展示给客户参观，能让客户对公司的其他产品或者更多的

产品进行了解。

- 友情链接：这个功能大家肯定不会陌生，它主要是网站与网站之间的相互宣传，同样该模块我们可以在后台进行管理。

2. 后台模块

- 内容管理：内容管理中分为栏目管理(导航管理)、添加内容、添加标示内容、内容管理和标示内容管理等功能，在该功能中，可以添加管理首页中显示的新闻信息、公司简介、联系方式、产品展示和案例展示等内容。
- 用户管理：在该模块中还有管理员管理、管理员分组、会员管理和会员分组等功能，可以对该功能内的信息进行添加、更改、删除等。
- 友情链接：在首页最下端有友情链接等功能，它的好处是有利于用户的扩展阅读，能更多的了解非本站以外的内容。

3. 会员模块

- 修改个人信息：会员用户可以通过会员登录窗口进入会员管理页面，在该页面中可以修改个人的手机号、QQ 号等信息。
- 咨询问题：同样会员还有一个功能是向网站管理人员进行咨询，当在使用过程中有不解的地方可以通过会员管理界面提交需要咨询的问题。
- 添加收藏：收藏功能是将在网站中发现对自己有用或者经常需要关注的文章或者其他信息进行收藏，方便以后的阅读。
- 修改个人密码：会员登录进入系统后，可以对自己的用户密码进行修改。

14.2 数据库设计

企业宣传网系统使用了 MySQL 数据库，设计数据库名字为 dbs，经过以上的系统分析后，可以把数据库中的表主要划分为 10 张表，分别是管理员用户表、管理员分组表、会员咨询表、文章内容表、标示内容表、站内导航表、招聘信息表、友情链接表、会员信息表和会员分组表。以下是表的详细结构信息。

1. 管理员用户表

用来保存管理员用户信息内容的表格，该表中有 11 个字段分别为用户 id、登录名、登录密码、昵称、分组 id、真实姓名、邮箱、电话、是否禁用、注册 IP 和注册时间等字段。管理员用户表信息如表 14-1 所示。

表 14-1　管理员用户表 bees_admin

字段名称	含　义	类　型	约　束
id	用户 id	mediumint	主键
admin_name	登录名	varchar(225)	非空
admin_password	登录密码	varchar(225)	非空
admin_nich	昵称	varchar(225)	非空

续表

字段名称	含　义	类　型	约　束
admin_purview	分组 id	mediumint	非空
admin_admin	真实姓名	varchar(255)	空
admin_mail	邮箱	varhcar(255)	空
admin_tel	电话	varhcar(255)	空
is_disable	是否禁用	mediumint	非空
admin_ip	注册 IP	varhcar(255)	空
admin_time	注册时间	varhcar(255)	空

2．管理员分组表

该表用来保存管理员分组信息，该表有 5 个字段分别为管理组 id、管理组名称、管理组描述、管理组权限和是否禁用等字段。表 14-2 则是管理员分组信息表。

表 14-2　管理员分组表 bees_admin_group

字段名称	含　义	类　型	约　束
id	管理组 id	mediumint	主键
admin_group_name	管理组名称	varchar(225)	非空
admin_group_info	管理组描述	varchar(225)	空
admin_group_purview	管理组权限	varchar(225)	非空
is_disable	是否禁用	mediumint	非空

3．文章内容表

文章内容表用来保存添加的产品描述内容信息。该表有两个字段分别为内容 id 和内容信息字段，该表信息如表 14-3 所示。

表 14-3　文章内容表 bees_article

字段名称	含　义	类　型	约　束
id	文章 id	mediumint	主键
content	text	varchar(225)	空

4．标示内容表

标示内容表中保存标示内容信息，比如首页中显示的公司简介或者联系方式等内容信息，该表中有 3 个字段分别为标示 id、标示标题和标示内容等字段，该表信息如表 14-4 所示。

表 14-4　表示内容表 bees_block

字段名称	含　义	类　型	约　束
id	标示 id	mediumint	主键
tag	标示标题	varchar(225)	非空
content	标示内容	text	空

5. 站内导航表

用来保存网站前台导航信息的表，该表共有 10 个字段，分别为导航内容 id、导航名称、是否新窗口打开、是否隐藏栏目、内容模型、栏目目录名称、排列顺序、导航显示、栏目使用页面和外链接转向页面等字段。详细表信息如表 14-5 所示。

表 14-5 站内导航表

字段名称	含 义	类 型	约 束
id	导航内容 id	mediumint	主键
cate_name	导航名称	varchar(225)	非空
cate_mb_is	是否新窗口打开	mediumint	非空
cate_hide	是否隐藏栏目	mediumint	非空
cate_channel	内容模型	mediumint	非空
cate_fold_name	栏目目录名称	varchar(225)	非空
cate_order	排列顺序	mediumint	非空
cate_rank	导航显示	mediumint	非空
cate_tpl	栏目使用页面	mediumint	非空
cate_url	外链接转向页面	varchar(225)	空

6. 招聘信息表

该表是用来保存招聘职位的信息内容，该表中有 16 个字段，分别为招聘信息 id、招聘类型、招聘人数、工作经验、工作地点、年龄、身高、语言及水平要求、截至日期、待遇、学历、性别、联系电话、邮箱、网址和详细内容等字段，该表详细信息表 14-6 所示。

表 14-6 招聘信息表 bees_job

字段名称	含 义	类 型	约 束
id	招聘信息 id	mediumint	主键
jobtype	招聘类型	varchar(225)	空
jobnum	招聘人数	varchar(225)	空
jobexp	工作经验	varchar(225)	空
jopaddress	工作地点	varchar(225)	空
jobage	年龄	varchar(225)	空
jobheight	身高	varchar(225)	空
joblanguage	语言及水平要求	varchar(225)	空
joblasttime	截止时间	varchar(225)	空
jobsalary	待遇	varchar(225)	空
jobedu	学历	varchar(225)	空
jobsex	性别	varchar(225)	空
jobtel	联系电话	varchar(225)	空

续表

字段名称	含　义	类　型	约　束
jobmail	邮箱	varchar(225)	空
jobweb	网址	varchar(225)	空
content	详细内容	text	空

7．友情链接表

该表中保存了首页底部显示的友情链接信息内容，通过对该表的操作可以增加或删除友情链接信息，表中共有 8 个字段分别为链接 id、网站网址、网站名称、网站 logo、显示顺序、网站说明、站长邮箱、添加时间等字段，该表详细信息如表 14-7 所示。

表 14-7　友情链接表 bees_link

字段名称	含　义	类　型	约　束
id	链接 id	mediumint	主键
link_url	网站网址	varchar(225)	非空
link_name	网站名称	varchar(225)	非空
link_logo	网站 logo	varchar(225)	空
link_order	显示顺序	mediumint	非空
link_info	网站说明	varchar(225)	空
link_mail	站长邮箱	varchar(225)	空
addtime	添加时间	varchar	空

8．会员信息表

会员信息表中保存了会员用户的基本信息，该表有 9 字段分别为会员 id、用户名、用户密码、用户昵称、用户组、登录名称、邮箱、电话和是否禁用等字段，该表信息如表 14-8 所示。

表 14-8　会员信息表 bees_member

字段名称	含　义	类　型	约　束
id	会员 id	mediumint	主键
member_name	用户名	varchar(225)	空
member_password	用户密码	varchar(225)	非空
member_nich	用户昵称	varchar(225)	非空
member_purview	用户组	varchar(225)	非空
member_user	登录名称	varchar(225)	非空
member_mail	邮箱	varchar(225)	非空
member_mail	电话	varchar(225)	空
is_disable	是否禁用	mediumint	非空

9. 会员分组表

保存了会员用户的类型，该表中有 4 个字段，着 4 个字段分别为分组 id、分组名称、描述、是否禁用等字段，该表详细信息如表 14-9 所示。

表 14-9 会员分组表

字段名称	含 义	类 型	约 束
id	分组 id	mediumint	主键
member_group_name	分组名称	varchar(225)	非空
member_group_info	描述	varchar(225)	非空
is_disable	是否禁用	mediumint	非空

10. 会员咨询

该表主要保存的是会员登录操作页面后提交的咨询内容信息，管理人员可以对此信息进行回复，该表共有 7 个字段，它们分别为标示 id、咨询标题、咨询内容、咨询时间、管理回复、咨询会员、回复时间等字段，该字段详细信息如表 14-10 所示。

表 14-10 会员资询表

字段名称	含 义	类 型	约 束
id	标示 id	mediumint	主键
title	咨询标题	varchar(225)	非空
content	咨询内容	text	非空
addtime	咨询时间	varchar(225)	非空
reply	管理回复	varchar(225)	空
member	咨询会员	mediumint	非空
replytime	回复时间	varchar(225)	空

14.3 系统运行环境

本节主要讲解的内容是开发环境和运行环境的搭建，在学习本节之前，先消除对学习的紧张感。其实学习是很轻松的事，学习并不神秘(如果搞得那么复杂，估计也不会发展那么迅速了)，同样电脑只是一种工具，电脑的内部工作原理虽然很复杂，但不需要了解那些深奥的原理，只要知道怎样使用电脑开发出我们想要的项目就可以了。在表 14-11 中则是搭建环境需要的软件，下载后进行安装即可。

表 14-11 环境配置

软 件	版 本	下载地址
MySQL	5.0	http://dev.mysql.com/
Apache	2.2	http://www.apache.org
Dreamweaver CS4	4.0	http://dl.pconline.com.cn/html_2/1/87/id=1175&pn=0.html

1. 硬件环境

- 处理器：Inte1Pentium
- 内存：32MB 或更高
- 硬盘空间：1GB

2. 软件环境

- 操作系统：Windows 2003 /2000/XP
- Web 服务器：Apache 2.2 或以上版本
- 数据库：MySQL
- 开发工具：Dreamweaver CS4
- 客户端：IE 6.0 或以上版本
- 开发语言：PHP

开发环境和软件安装配置好之后，使用 Dreamweaver CS4 创建新的站点命名为“enterprise”，图 14-4 是企业宣传网的结构图。

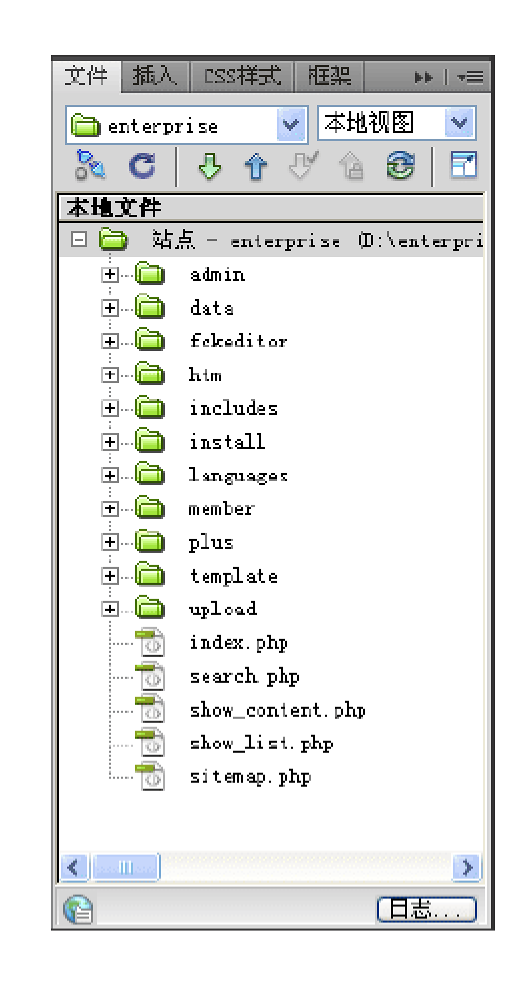


图 14-4　企业宣传网结构图

14.4　连接数据库

大多数企业宣传网站中，首页显示的数据信息或者后台管理的数据都将会保存至数据库中，同样，我们编写的企业宣传网站也有大量数据需要处理，因此在编写程序时就需要和数据库进行交互，在对数据库交互之前，首先需要打开数据库连接，为了方便今后对系统的维护，将连接数据库的用户信息和数据库名封装到 confing.php 类中，代码如下所示：

```
<?php
//连接服务器名称
define('DB_HOST', 'localhost');
//连接数据库账号
define('DB_USER', 'root');
//连接数据库密码
define('DB_PASSWORD','root');
//连接数据库名
define('DB_NAME', 'dbs');
define('DB_PRE', 'bees_');
define('DB_PCONNECT', 0);
//连接数据库编码格式
define('DB_CHARSET', 'utf8');
define('CMS_ADDTIME',1295256714);
?>
```

在代码中通过使用 define()方法保存连接数据里的一些登录名、登录密码、服务器名称和连接数据库名称等信息，接下来是进行连接数据库，同样，将连接数据库信息也保存到一个名为 mysql.class.php 的公共类中，代码如下所示：

```
    var $link;
    var $db_host;
    var $db_user;
    var $db_password;
    var $db;
    var $db_prefix;
    var $db_pc;
function mysql($db_host,$db_user,$db_password,$db,$db_charset,$pc_connect){
        $this->db_host=$db_host;              //获取服务器名
        $this->db_user=$db_user;              //获取登录账号
        $this->db_password=$db_password;      //获取登录密码
        $this->db=$db;                        //获取数据库名
        $this->db_charset=$db_charset;        //编码格式
        $this->db_pc=$pc_connect;
        $this->open_link();
        $this->sl_db();
    }
```

在以上代码中是在该类中获取 confing.php 类中定义的一些连接数据库用户名和用户密码等信息内容，保存到声明好的全局变量中。

接下来的代码则是对数据库进行连接的业务操作，首先创建了 open_link()的方法用来连接数据库。

```
    //连接数据库
    function open_link(){
    $this->link=($this->db_pc)?@mysql_pconnect($this->db_host,$this->db_user
,$this->db_password):@mysql_connect($this->db_host,$this->db_user,$this->db_
password);
        if(!$this->link){
            err('数据库连接错误'.mysql_error(),"javascript:history.go(-1);");
        }
    }
    //查询数据库是否存在
    function sl_db(){
        if(!@mysql_select_db($this->db,$this->link)){
            err('数据库'.$this->db.'不存在',"javascript:history.go(-1);");
        }
        @mysql_unbuffered_query("set names {$this->db_charset}");
    }
    //判断查询操作是否成功
    function query($sql){
        if(!$res=@mysql_query($sql,$this->link)){
            err('操作数据库失败'.mysql_error()."<br>sql:{$sql}","javascript:
history.go(-1);");
        }
        return $res;
    }
```

在上述代码中首先将 mysql_pconnect()方法进行连接数据库操作将返回信息保存至声明好的 link 变量中，然后进行判断连接数据库是否成功，如果错误则返回连接数据库请求之前的页面中；如果连接成功，执行以下自定义 sl_db()方法，该方法中使用 mysql_select_db()来查询需要请求的数据库是否存在，如果存在则执行自定义 query()方法。该方法有一个参数用来专递执行的 SQL 语句使用 mysql_query()来执行 SQL 查询，最后判断查询数据是否成功，如果成功则返回结果集。

为了方便以后开发使用，避免查询数据库内容信息时反复编写同样的代码，我们将有关查询的所有方法也封装到该类中，以下代码则是使用 SQL 语句执行查询取出数字作为数组索引的数据集合代码。

```
/*
    *取出数字作为数组索引的数据集合
    *return $array
    */
    function fetch_array($sql){
        $result=$this->query($sql);
        while($rows=mysql_fetch_array($result)){
            $arr[]=$rows;
        }
        mysql_free_result($result);
        return $arr;
    }

    function fetch_asc_list($sql){
        $result=$this->query($sql);
        return mysql_fetch_assoc($result);
    }
```

在以上代码中首先创建了自定义 fetch_array()这样一个方法，该方法有一个参数，是用来传入 SQL 语句，在该方法中，调用之前创建好的自定义 query()方法对数据库进行查询并返回 res 结果集。然后使用 while 循环，循环条件使用 mysql_fetch_array()方法遍历结果集保存到数组中返回。代码中自定义了 fetch_asc_list()方法用来返回一个集合行书结果集。

```
/*
    *取得上次添加数据的 id
    *return $id
    */
    function insert_id(){
        return mysql_insert_id($this->link);
    }
    //取得一行数据
    function get_row($sql,$row=0,$field=0){
        $rel=$this->query($sql);
        $row=mysql_result($rel,$row,$field);
        mysql_free_result($rel);
        return $row;
    }
```

```
    //返回数据数目
    function fetch_rows($sql){
        $result=$this->query($sql);
        $num=@mysql_num_rows($result);
        mysql_free_result($result);
        return $num;
    }
    //返回字段名数组
    function fetch_field($sql){
        $result=$this->query($sql);
        $num=mysql_num_fields($result);
        for($i=0;$i<$num;$i++){
            $arr[]=mysql_field_name($result,$i);
        }
        return $arr;
    }
```

在代码中自定义 insert_id()方法用来获取上次添加数据 id，在该方法中使用 mysql_insert_id()来获取，mysql_insert_id()方法有一个参数是数据库连接语句。代码中 get_row()方法用来返回一条数据，经常用到查询详细信息的操作，fetch_rows()方法则是用来返回该 SQL 语句的数据条数，方法 fetch_field()用来以数组形式返回字段名。

```
    //创建数据表
    function create_tb($table,$field){
        $sql="create table ".DB_PRE.$table." (".$field.") ENGINE = MYISAM
CHARACTER SET utf8 COLLATE utf8_general_ci";
        return $this->query($sql);
    }
    //添加字段
    function add_field($table,$field){
        $sql="alter table ".DB_PRE.$table." add ".$field." CHARACTER SET utf8
COLLATE utf8_general_ci NULL";
        return $this->query($sql);
    }
    //修改字段
    function xg_field($table,$field){
        $sql="alter table ".DB_PRE.$table." CHANGE ".$field." CHARACTER SET
utf8 COLLATE utf8_general_ci NULL DEFAULT NULL";
        return $this->query($sql);
    }
    //删除字段
    function drop_field($table,$field){
        $sql="alter table ".DB_PRE.$table." drop column ".$field;
        return $this->query($sql);
    }
```

代码中，自定义了 4 个方法分别用来创建数据库表、插入字段、修改字段和删除字段等功能，之所以将这些公共方法提取出来写的原因是方便以后经常使用这些方法书写重复的代码，这样不是编程的真正思想，首先代码中自定义 create_tb()方法是用来创建一个数据表，该方法

有两个参数，自定义 add_field()方法是用来插入字段，同样该方法有两个参数，自定义 xg_field()方法用来修改字段，drop_field()方法则是用来删除字段的操作。

14.5　用 户 登 录

对于大多数系统来说，都会有登录这样的功能，同样我们做企业宣传网也有同样的功能，在这里我们的登录有两种，一种是前台的会员登录，另一种则是后台的管理员登录，而会员登录窗口还可以进行会员注册功能，因此本节将借助登录和会员注册进行讲解。

14.5.1　会员登录及注册

1. 会员注册

要想成为该网站的会员首先必须通过申请用户信息才可以成为本网站会员用户，才能享用会员的一些操作功能，注册网站会员首先需要填写用户名、昵称、登录密码、确认密码、电子邮箱和验证码等信息，详细如图 14-5 所示。



图 14-5　会员注册

用户在注册页面输入用户信息后单击【确定】按钮进行提交注册，在注册之前将会对用户输入的用户信息进行验证是否为空和一些密码、邮箱格式验证，通过验证执行插入数据功能，详细代码如下所示：

```
$user=fl_html(fl_value($_POST['user']));
$password=fl_html(fl_value($_POST['password']));
$password2=fl_html(fl_value($_POST['password2']));
$nich=fl_html(fl_value($_POST['nich']));
$mail=fl_html(fl_value($_POST['mail']));
$code=fl_html(fl_value($_POST['code']));
if(!$_sys['web_member'][0]){
    die("<script  type=\"text/javascript\">alert(' 会 员 注 册 已 经 暂 停
```

```
');history.go(-1);</script>");
    }
    if(!check_str($user,'^[a-zA-Z][a-zA-Z0-9]{3,15}$')){die("<script
type=\"text/javascript\">alert('用户名只能是字母数字，4 以上长度');history.go(-1);
</script>");}
    if(!check_str($nich,'^[a-zA-Z][a-zA-Z0-9]{3,15}$')){die("<script type=\"text/
javascript\">alert('昵称只能是字母数字，4 以上长度');history.go(-1);</script>");}
    if(empty($password)||empty($password2)){die("<script
type=\"text/javascript\">alert('密码不能为空');history.go(-1);</script>");}
    if($password!=$password2){die("<script  type=\"text/javascript\">alert('
两次密码不一样');history.go(-1);</script>");}
    if(!check_str($mail,'^[0-9a-z]+@(([0-9a-z]+)[.])+[a-z]{2,3}$')){die("<s
cript type=\"text/javascript\">alert('邮箱不正确');history.go(-1);</script>");}
    if(!empty($_sys['member_no_name'])){$no_name=explode('|',$_sys['member_n
o_name']);}
    if(is_array($no_name)){
        if(in_array($user,$no_name)){die("<script
type=\"text/javascript\">alert(' 【 ".$user." 】 该 用 户 名 不 能 注 册
');history.go(-1);</script>");}
    }
```

上述代码中，是对用户提交的用户信息通过正则进行基本验证，首先判断会员注册功能是否开通，如果开通则可以进行注册，判断用户名为字母组成，而且不能少于 4 位，其次判断昵称，接着判断密码不为空和两次输入密码是否相同，最后则判断邮箱格式是否正确，还有一项判断则是用来判断用户名是否存在恶意性的词语进行过滤，用户提交的信息已经验证完成了，接下来则是验证验证码输入是否正确，代码如下所示：

```
    if(!empty($_sys['safe_open'])){
        foreach($_sys['safe_open'] as $k=>$v){
        if($v=='1'){
            if($code!=$_SESSION['code']){die("<script
type=\"text/javascript\">alert('验证码不正确');history.go(-1);</script>");}
        }
        }
        }
```

在上述代码中，首先获取用户输入的验证码，接下来和系统生成的验证码进行比较，如果相同则通过验证，否则提示验证码输入不正确，转向上一个页面。

在用户进行注册时经常会出现两个用户同时使用一个用户名进行注册，为了防止两个用户注册一个用户名的冲突，在注册前先进行查询数据库是否存在此用户，如果存在则提示用户名已经存在，详细代码如下所示：

```
    $sql="select id from ".DB_PRE."member where member_user='{$user}'";
    if($GLOBALS['mysql']->fetch_rows($sql)){die('已经存在相同的用户名，请更换用户
    名');}
    if(!$_sys['member_mail'][0]){
    $sql="select id from ".DB_PRE."member where member_mail='{$mail}'";
    if($GLOBALS['mysql']->fetch_rows($sql)){die($mail.'该邮箱已经被使用，请更换
```

```
');}
  }
```

系统中，为了防止邮箱的冲突，在注册验证之前也对邮箱进行了查询，是否存在此邮箱被注册的过程。

当以上信息通过了验证之后，则可以进行向数据库进行插入数据，将用户提交的信息插入数据库内，代码如下所示：

```
    $addtime=time();
    $password=md5($password);
    $sql="insert into ".DB_PRE."member
(member_user,member_password,member_nich,member_mail,member_purview)
 values ('{$user}','{$password}','{$nich}','{$mail}',1)";
    $GLOBALS['mysql']->query($sql);
    $last_id=$GLOBALS['mysql']->insert_id();
    $ip=fl_html(fl_value(get_ip()));
    $sql="update  ".DB_PRE."member  set  member_time='{$addtime}',member_ip=
'{$ip}' where id={$last_id}";
    $GLOBALS['mysql']->query($sql);
    echo "<script type=\"text/javascript\">alert('用户注册成功');location.
href='member.php';</script>";
```

上述代码中获取当前的系统时间、md5 加密后的字符串和之前验证好的用户基本信息进行插入，首先拼接 SQL 字符串，调用 query()方法进行插入，插入成功之后获取当前用户 IP 地址，将数据库 IP 地址修改为当前用户 IP 地址，最后提示用户注册成功，转向登录页面。

2. 会员登录

普通用户要想使用向管理员提意见功能和收藏功能，必须进行登录，只有登录后的会员才拥有以上功能，因此就设计了会员登录的功能，会员登录时需要输入用户名、登录密码和验证码 3 项内容，如图 14-6 所示。

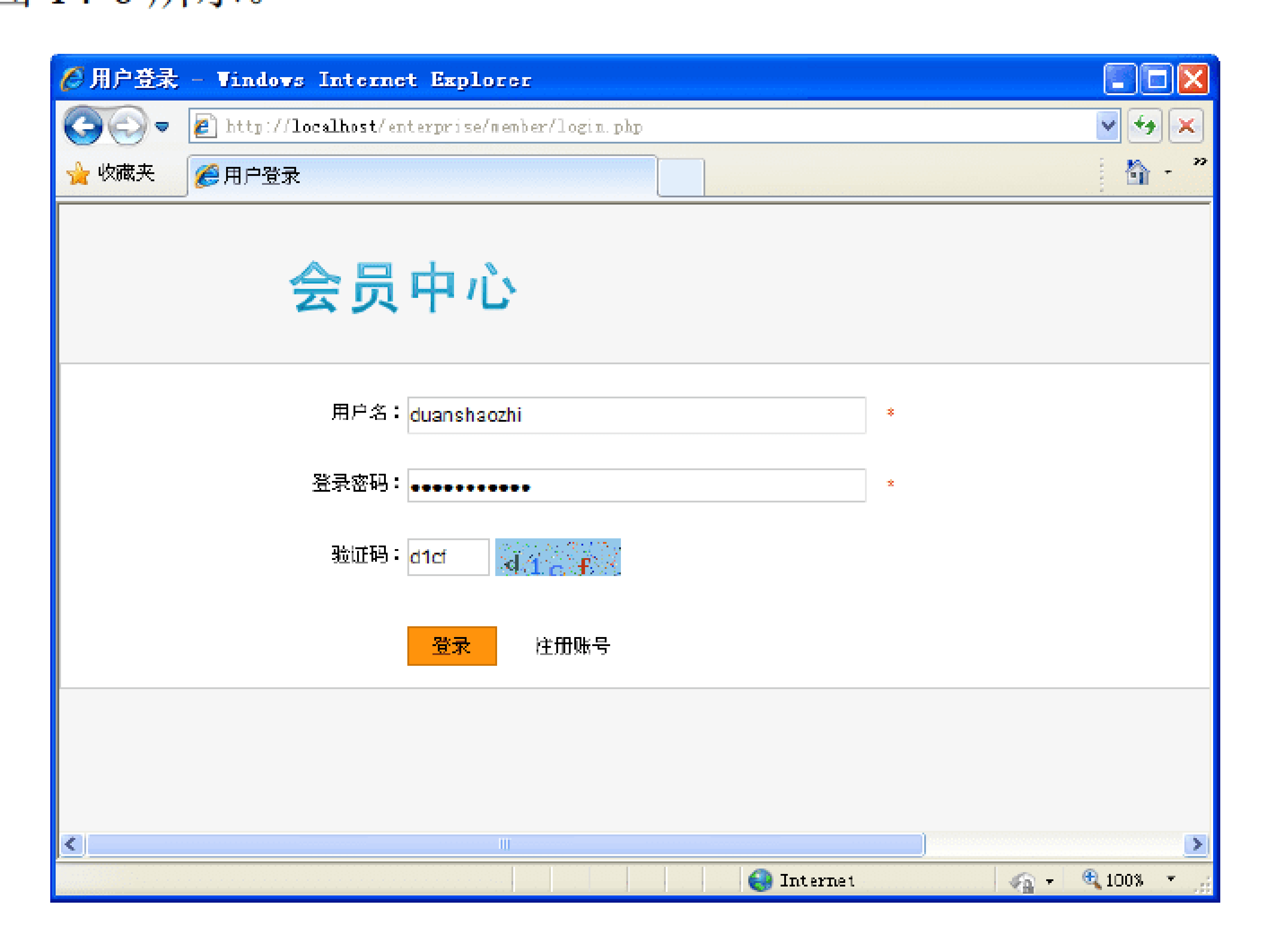


图 14-6　会员登录

说起来很简单，但是做起来还是需要用心的，接下来就使用 PHP 技术来实现会员登录的功能，前期已经将数据库连接和操作类编写好了，现在只需在要对数据库进行操作的类中引用该类即可，下面先看代码：

```
function save_login(){
    global $submit,$user,$password,$_sys,$code;
    $submit=$_POST['submit'];
    $user=fl_html(fl_value($_POST['user']));
    $password=fl_html(fl_value($_POST['password']));
    $code=$_POST['code'];
    if(empty($submit)){die("<script type=\"text/javascript\">alert('请先
        登录');location.href='login.php';</script>");}
        if(!empty($_sys['safe_open'])){
        foreach($_sys['safe_open'] as $k=>$v){
        if($v=='2'){
            if($code!=$_SESSION['code']){die("<script
type=\"text/javascript\">alert('验证码不正确');history.go(-1);</script>");}
        }
        }
        }
    if(empty($user)||empty($password)){die("<script type=\"text/javascript\">
        alert('用户名或密码不能为空');history.go(-1);</script>");}
    $password=md5($password);
    $sql="select*from ".DB_PRE."member where member_user='{$user}' and member_
        password='{$password}'";
    if(!$GLOBALS['mysql']->fetch_rows($sql)){die("<script type=\"text/javascript\">
        alert('用户名或密码不正确');history.go(-1);</script>");}
    $rel=$GLOBALS['mysql']->fetch_asc($sql);
    if($rel[0]['is_disable']){die("<script type=\"text/javascript\">alert('
        登录失败，该账号已被锁定');history.go(-1);</script>");}
    $_SESSION['member_user']=$rel[0]['member_user'];
    $_SESSION['login']='true';
    $_SESSION['id']=$rel[0]['id'];
    $_SESSION['purview']=$rel[0]['member_purview'];
    $m_time=time();
    $_SESSION['m_time']=$m_time;
    $m_ip=fl_html(fl_value(get_ip()));
    $_SESSION['m_ip']=$m_ip;
    $m_count=$rel[0]['member_count']+1;
    $sql="update ".DB_PRE."member set member_count={$m_count} where id= ".
        $rel[0]['id'];
    unset($rel);
    $GLOBALS['mysql']->query($sql);
    header("location:member.php");
}
```

代码中，自定义了 save_login()方法进行登录，首先声明了 5 个全局变量，用来保存提交信息、用户名、密码、验证码等信息。首先需要验证用户输入的验证码是否正确，如果正确则

执行用户名密码为空的验证，通过验证后判断用户密码是否正确，调用 md5()方法将用户输入密码进行 md5 加密后如果 md5 后面有括号则用小写，其他用大写和数据库内进行对比，如果返回信息为空则用户不存在或者已经被锁定。登录成功后将用户基本信息保存至 Session 中转向会员管理页面。

14.5.2　后台管理员登录

后台管理登录界面是管理员登录后台管理系统的必经之路。该登录账号密码是不可以进行注册的管理员内部指定的值，因此只有管理人员才可以进入后台系统，要想通过后台登录必须输入用户名、密码和验证码等信息，默认用户名为“admin”、密码为“admin”，详细如图 14-7 所示。



图 14-7　后台登录

同样，为了避免一些不合法的登录方式，在管理登录页面也进行了登录信息的基本验证，验证输入用户名和密码是否正确、是否为空的操作，然后在进行查询避免查询数据库时给服务器压力过大，详细代码如下所示：

```
function ck_login(){
    global $submit,$user,$password,$_sys,$code;
    $submit=$_POST['submit'];
    $user=fl_html(fl_value($_POST['user']));
    $password=fl_html(fl_value($_POST['password']));
    $code=$_POST['code'];
    if(!isset($submit)){
        msg('请从登录页面进入');
    }
    if(empty($user)||empty($password)){
        msg("密码或用户名不能为空");
    }
    if(!empty($_sys['safe_open'])){
```

```
        foreach($_sys['safe_open'] as $k=>$v){
        if($v=='3'){
            if($code!=$_SESSION['code']){msg("验证码不正确! ");}
        }
        }
        }
    check_login($user,$password);

}
```

代码中，定义全局变量，获取表单提交过来的数据信息，将用户名、用户密码和验证码保存至全局变量中进行判断，如果用户提交信息不是从登录页面提交则提示用户到登录页面进行提交，接下来则是判断用户输入用户名是否为空，如果不为空则进行下一个判断，最后则是验证用户输入验证码是否正确，使用 foreach 循环遍历进行判断。如果验证都通过则调用自定义函数 check_login()，该函数代码如下所示：

```
function check_login($user,$password){
    $rel=$GLOBALS['mysql']->fetch_asc("select
id,admin_name,admin_password,admin_purview,is_disable
 from ".DB_PRE."admin where admin_name='".$user."' limit 0,1");
    $rel=empty($rel)?'':$rel[0];
    if(empty($rel)){
        msg('不存在该管理用户','login.php');
    }
    $password=md5($password);
    if($password!=$rel['admin_password']){
        msg("输入的密码不正确");
    }
    if($rel['is_disable']){
        msg('该账号已经被锁定,无法登录');
    }
    if (!isset($_SESSION)) {session_start();}
    $_SESSION['admin']=$rel['admin_name'];
    $_SESSION['admin_purview']=$rel['admin_purview'];
    $_SESSION['admin_id']=$rel['id'];
    $_SESSION['admin_time']=time();
    $_SESSION['login_in']=1;
    $_SESSION['login_time']=mktime();
    $ip=fl_value(get_ip());
    $ip=fl_html($ip);
    $_SESSION['admin_ip']=$ip;
    unset($rel);
    header("location:admin.php");
}
```

该方法中有两个参数用户名和密码，将用户名和密码拼接至 SQL 语句中进行查询，首先使用 empty()方法判断该用户输入用户名是否存在，如果存在则对密码进行 md5 加密，判断用户密码是否正确，然后判断用户账号是否被锁定，如果用户信息输入正常，则登录成功，将用户信息保存至 Session 中，同时记录登录用户的 IP 地址，转向管理界面。

14.6　会 员 管 理

在普通用户经过注册并注册成功之后通过会员登录窗口进行登录，通过所有的登录验证之后才可以到达会员管理界面，在登录界面中会显示当前的系统时间，在页面的左侧有中心首页、用户信息、会员咨询、我的收藏和修改密码等功能，本节使用 PHP 技术实现这些功能的应用，运行结果如图 14-8 所示。



图 14-8　会员中心

14.6.1　修改会员信息

修改用户信息的功能在大多数拥有会员功能的网站都会出现，因为这是作为一个普通用户必须拥有的功能，因此在这里凡是登录成功的用户登录该管理界面都可以进行修改用户基本信息的操作，例如用户的出生日期、性别、邮箱、联系 QQ、联系电话等信息，详细如图 14-9 所示。



图 14-9　用户信息修改

在执行该页面时首先需要判断用户是否登录，如果通过不正当的行为进入该页面将是不会进行修改的，在显示该页面时需要进行查询，将当前用户的基本信息显示在页面中，然后进行修改，详细代码如下所示：

```
if(empty($_SESSION['member_user'])||empty($_SESSION['id'])||!$_SESSION['log
in']){die("<script type=\"text/javascript\">alert('请先登录');location.href=
'login.php';</script>");}
$action=isset($action)?$action:'info';
go_url($action);
function info(){
    $sql="select*from ".DB_PRE."member where id=".$_SESSION['id'];
    $rel=$GLOBALS['mysql']->fetch_asc($sql);
    if(!empty($rel[0]['member_birth'])){$arr=explode('-',$rel[0]['member_bi
rth']);}
    include('template/member_info.html');
}
```

在代码中，首先获取 Session 中的用户信息进行判断是否 Session 存在用户信息，如果不存在则提示用户进行登录，如果存在则通过登录的用户 id 查询用户基本信息，执行以下 info()函数进行查询，将数据保存到变量中，转向 member_info.html 模板页面进行显示。

将用户之前的基本信息显示在页面后，用户可以对自己的基本信息进行修改，修改好基本信息进行提交时，将会通过一些基本信息的验证，通过验证后才可以修改成功，详细代码如下所示：

```
function save_info(){
    global
$birthdayYear,$birthdayMonth,$birthdayDay,$sex,$mail,$qq,$tel,$phone,$submit;
    $birthdayYear=fl_html(fl_value($_POST['birthdayYear']));
    $birthdayMonth=fl_html(fl_value($_POST['birthdayMonth']));
    $birthdayDay=fl_html(fl_value($_POST['birthdayDay']));
    $sex=fl_value($_POST['sex']);
    $mail=fl_html(fl_value($_POST['mail']));
    $qq=fl_html(fl_value($_POST['qq']));
    $tel=fl_html(fl_value($_POST['tel']));
    $phone=fl_html(fl_value($_POST['phone']));
    $submit=$_POST['submit'];
    if(!empty($qq)){
    if(!check_str($qq,'^[1-9][0-9]*$')){die("<script type=\"text/javascript\">
        alert('QQ 号码不正确');history.go(-1);</script>");}
    }
    if(!empty($phone)){
    if(!check_str($phone,'^[1-9][0-9]*$')){die("<script type=\"text/javascript\">
        alert('手机必须为数字');history.go(-1);</script>");}
    }
    if(empty($submit)){die("<script type=\"text/javascript\">alert(
        '请从表单提交');history.go(-1);</script>");}
    $birth=$birthdayYear.'-'.$birthdayMonth.'-'.$birthdayDay;
    $sql="update ".DB_PRE."member set member_tel='{$tel}',
```

```
member_phone='{$phone}',member_birth='{$birth}',member_sex
=".fl_html($sex[0]).",member_qq='{$qq}' where id={$_SESSION['id']}";
    $GLOBALS['mysql']->query($sql);
    echo  "<script  type=\"text/javascript\">alert(' 修 改 成 功 ');history.go
(-1);</script>";
}
```

在上述代码中，首先定义用来存储用户基本信息的全局变量，将用户请求的基本信息进行保存，对用户输入的 QQ 号码进行正则表达式验证，如果输入正确，则进行手心号码验证，如果用户信息输入全部正确则提交用户请求，将用户提交信息和 SQL 语句拼接之后调用 query()方法，将 SQL 语句传入该方法中进行修改数据操作，修改成功则给用户以提示。

14.6.2　会员咨询

同样，该功能也只能在会员登录成功后进行操作，会员可以在会员咨询模块添加咨询信息，在咨询信息添加页面需要输入咨询信息标题和咨询内容，详细如图 14-10 所示。



图 14-10　添加咨询信息

添加咨询信息，首先需要对提交数据进行判断是否为空或者输入内容是否合法，通过判断执行插入操作，详细代码如下所示：

```
function save_ask(){
    $member_id=intval($_POST['member_id']);
    $title=fl_html(fl_value($_POST['title']));
    $content=fl_html(fl_value($_POST['content']));
    if(empty($member_id)){die("参数传递错误,请重新操作");}
    if(empty($title)||empty($content)){die("标题或内容不能为空");}
    $addtime=time();
    $sql="insert into ".DB_PRE."ask
    (title,content,addtime,member) values ('{$title}','{$content}',
    '{$addtime}',{$member_id})";
    $GLOBALS['mysql']->query($sql);
    echo "<script type=\"text/javascript\">alert('咨询添加成功');history.go(-1);
```

```
</script>";
}
```

以上代码同样获取添加数据内容，将咨询标题、咨询问题和咨询人信息保存至 SQL 语句中调用 query()方法进行插入，插入成功提示用户咨询添加成功。

咨询信息添加成功之后可以对咨询问题信息进行查询，在转向咨询问题信息列表时需要进行查询，将当前登录用户咨询的问题信息显示在列表中，并且可以对咨询问题进行修改，但是要注意的是，如果管理员已经对其咨询问题进行回复，将不可以修改，运行结果如图 14-11 所示。



图 14-11　查询咨询问题

在显示当前用户所有咨询信息列表时，需要进行判断。如果已经被管理员进行回复的信息将状态显示为已回复，而且用红色高亮显示，否则显示未回复，详细代码如下所示：

```
function ask(){
    global $member_id;
    $member_id=$_SESSION['id'];
    $sql="select*from ".DB_PRE."ask where member=".intval($member_id)." order
by addtime desc";
    $rel=$GLOBALS['mysql']->fetch_asc($sql);
    include('template/member_ask.html');
}
```

代码中，获取 Session 中当前登录用户 id，通过 id 查询当前用户咨询所有问题信息，调用 fetch_asc()方法进行查询并且返回至页面中。

同样用户还可以对自己提交的咨询问题进行修改，在单击【修改】按钮时，将进行判断该信息是否被回复，如果没有回复则进入修改页面，将当前需要修改问题的基本信息显示在内容框中，运行结果如图 14-12 所示。



图 14-12　修改咨询问题

在修改咨询问题时，咨询问题的标题是不可以修改的，但可以对其咨询内容进行修改，在修改之前也需要通过一些基本信息验证，如果通过验证则可以进行修改操作，代码如下所示：

```
function save_xg(){
    global $id,$member_id,$content;
    $id=intval(fl_value($_POST['id']));
    $member_id=intval(fl_value($_POST['member_id']));
    $content=fl_html(fl_value($_POST['content']));
    if(empty($id)||empty($member_id)){die('参数传递错误,请重新操作');}
    if(empty($content)){die('内容不能为空');}
    $sql="update ".DB_PRE."ask set content='{$content}' where id={$id}";
    $GLOBALS['mysql']->query($sql);
    echo "<script type=\"text/javascript\">alert('咨询修改成功');location.href=
'?';</script>";
}
```

在上述代码中，首先定义两个全局变量来保存修改信息内容，接下来是判断接收到 id 是否为空，如果不为空则执行判断输入内容是否为空，不为空则进行 SQL 语句拼接，将修改信息内容拼接起来，调用 query()方法将 SQL 语句传入进去执行修改操作，修改成功则提示用户。

14.6.3　修改密码

修改密码功能也是大家再熟悉不过的一项功能，之前经常提到的 QQ、博客等系统都可以对自己用户密码修改，这样可以提升用户信息的安全性，防止密码被盗取，企业宣传网会员系统同样具备这样的功能，该功能的实现思路类似修改用户信息，在我们修改用户密码页面中会发现有旧密码、新密码和确认新密码 3 个文本框，之所以需要输入两次密码的原因是防止用户输入密码时输入错误而导致的不必要错误，该功能运行结果如图 14-13 所示。



图 14-13　修改密码

修改用户密码之前需要对当前用户密码进行验证，只有通过验证后的才可以进入修改密码阶段，修改密码时需要输入两次密码，确保用户输入密码无误，并且对输入密码进行基本的验证，然后提交修改信息，详细代码如下所示：

```
function save_password(){
    global $password_use,$password_new,$password_new2;
    $password_use=trim(fl_html(fl_value($_POST['password_use'])));
    $password_new=trim(fl_html(fl_value($_POST['password_new'])));
    $password_new2=trim(fl_html(fl_value($_POST['password_new2'])));
    if(empty($password_use)||empty($password_new)||empty($password_new2)){die("<script type=\"text/javascript\">alert('密码不能为空');history.go(-1);</script>");}
    $sql="select member_password from ".DB_PRE."member where id=".$_SESSION['id'];
    $rel=$GLOBALS['mysql']->get_row($sql);
    if(md5($password_use)!=$rel){die("<script type=\"text/javascript\">alert('原始密码不正确');history.go(-1);</script>");}
    if($password_new!=$password_new2){die("<script type=\"text/javascript\">alert('两次新密码不一样');history.go(-1);</script>");}
    $sql="update ".DB_PRE."member set member_password='".md5 $password_new)."' where id=".$_SESSION['id'];
    $GLOBALS['mysql']->query($sql);
    echo "<script type=\"text/javascript\">alert('密码修改成功');history.go(-1);</script>";
}
```

从 save_password()方法中首先定义用户当前密码、用户新密码和用户确认密码 3 个全局变量，然后使用 empty()方法进行判断用户输入信息是否为空，如果不为空则查询用户输入的当前密码是否正确，如果用户输入的密码通过验证则将输入的新密码进行 md5 加密，并用拼接字符串调用 query()方法进行修改，修改成功则提示用户。

14.7　内 容 管 理

本节讲解的内容都是在管理员登录成功后的一些操作功能，普通用户和会员是无法对此进行操作的，内容管理模板包括有栏目管理(导航管理)、添加内容、添加单页内容、添加标示内容、内容管理、单页内容管理、标示内容管理和上传附件管理等几项功能，因为在内容管理模块中，有很多功能实现是非常相似的，在这里只针对部分功能进行讲解。

14.7.1　添加标示内容

该功能用来添加标示内容(只包含内容)，内容形式如首页简介、联系方式等页面部分需要手工修改的部分信息，该功能的表示内容使用了 FCKeditor 工具，通过使用该工具可以让我们显示的内容加上修饰和样式等功能，也方便对显示内容的布局和排版，该功能运行结果如图 14-14 所示。

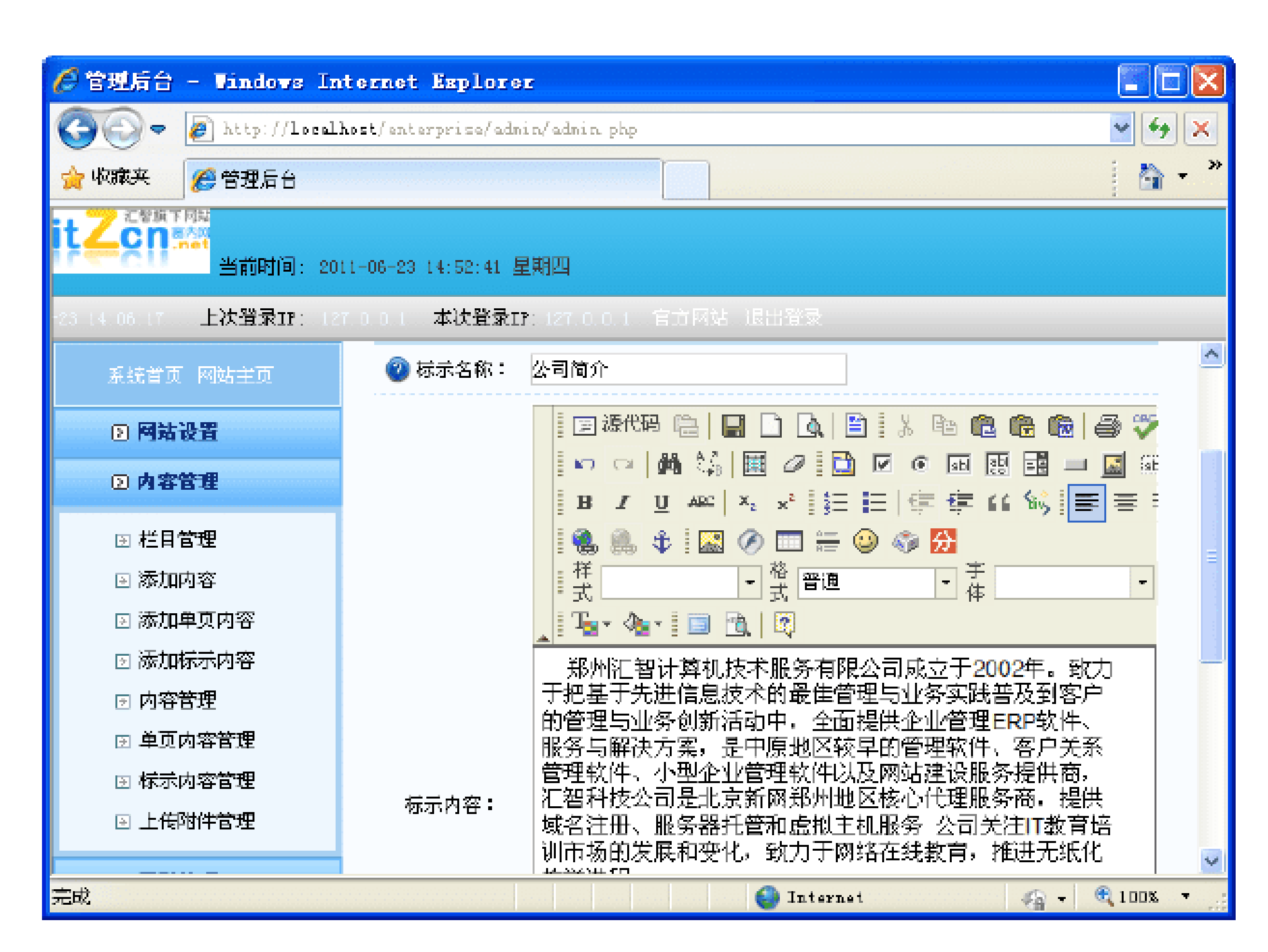


图 14-14　添加标示内容

同样在添加标示内容时也需要对用户输入的标题信息进行不能为空或含有其他字符等验证，但是标示内容是可以为空的，确认用户输入信息无误则进行添加操作，详细代码如下所示：

```
function save_content(){
    if(!check_purview('content_create')){msg('<span style="color:red">操作失败,你的权限不足!</span>');}
    global $tag,$content,$lang;
    if(empty($tag)){
        msg('<span style="color:red">标示名称不能为空</span>');
    }
    $tag_num=$GLOBALS['mysql']->fetch_rows("select id from ".DB_PRE."block where tag='{$tag}'");
    if($tag_num){msg('<span style="color:red">【'.$tag.'】内容已经存在，请更换</span>');}
```

```
    if(file_exists(DATA_PATH."sys_info.php")){include(DATA_PATH."sys_info.php");}
   $tag=empty($_sys['web_content_title'])?cn_substr($tag,60):cn_substr($tag,
        intval($_sys['web_content_title']));
    //关键字设置
    $key_arr=$GLOBALS['mysql']->fetch_asc('select*from '.DB_PRE."keywords where
        lang='".$lang."'");
    if(!empty($key_arr)){
        foreach($key_arr as $key_k=>$key_v){
        $content=str_replace($key_v['keywords'],'<a href="'.$key_v
            ['wordsurl'].'">'.$key_v['keywords'].'</a>',$content);
        }
    }
    $sql="insert into ".DB_PRE."block (tag,content) values ('{$tag}','{$content}')";
    $GLOBALS['mysql']->query($sql);
    msg("【{$tag}】内容添加完成");
}
```

在实现该功能代码时，首先判断当前登录用户是否有权限来操作该功能，在代码中判断用户输入标题是否为空，如果为空则提示用户标题名称不能为空的信息，否则执行查询该标题是否在数据库内存在，实现该功能的作用是严禁重复发布同样标题的标示内容信息，接下来则是获取 FCKeditor 中的内容，对关键字进行设置，最后则是将用户输入信息拼接到 SQL 语句调用 query()方法中执行插入操作，提示用户插入成功的信息。

14.7.2 标示内容管理

标示内容管理模块则是针对添加的标示内容信息进行同意管理和分配，在该模块中，可以对已经添加好的标示内容进行修改和删除的操作，运行结果如图 14-15 所示。

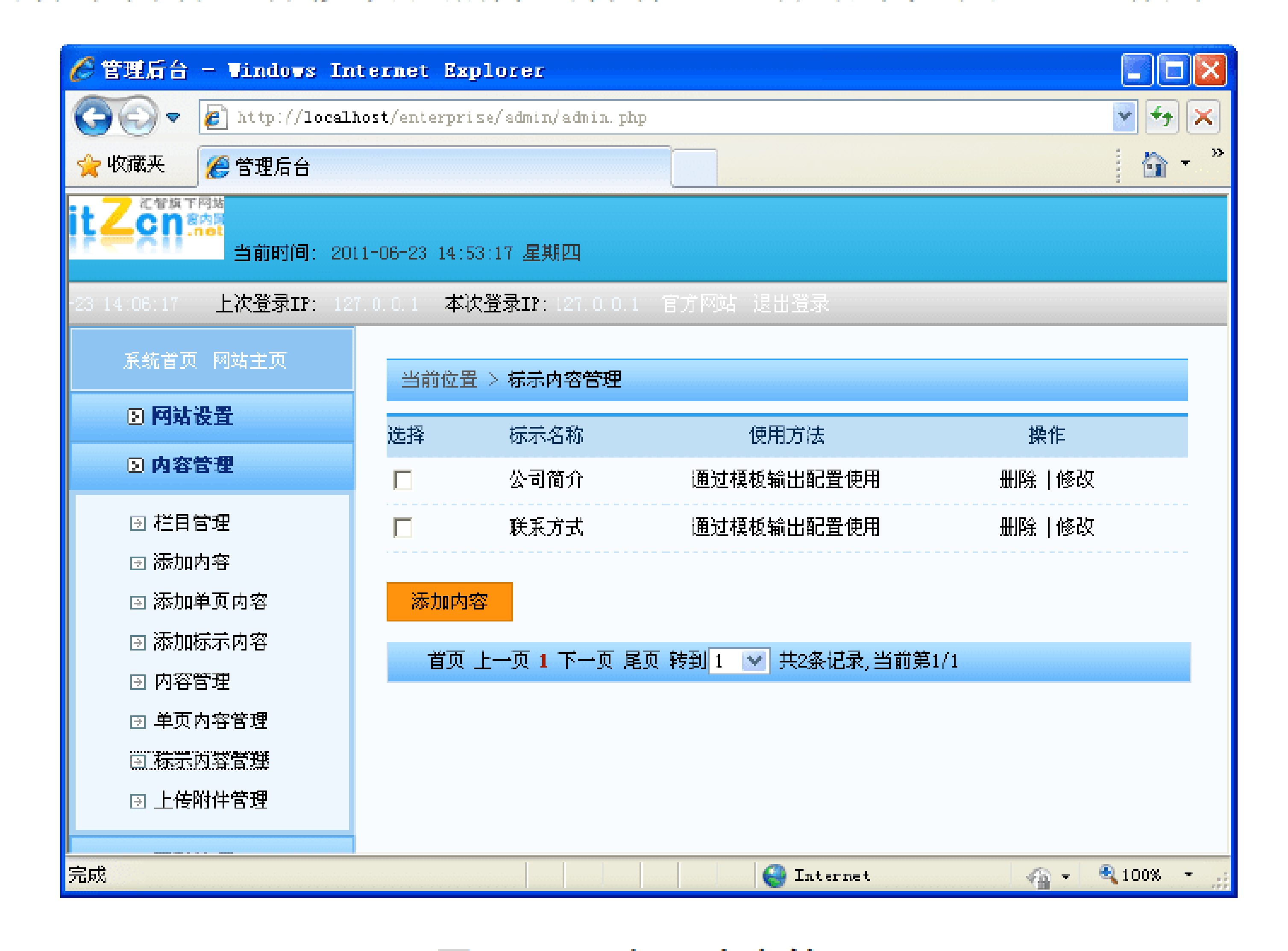


图 14-15 标示内容管理

在该功能模块中，首先将数据库中的所有标示内容通过列表形式显示在页面中，并且实现分页效果，当内容过多或者超过显示限制时则显示在第二页中，接下来可以将标示内容进行修改，当用户单击【修改】按钮时转向修改页面，如图 14-16 所示。

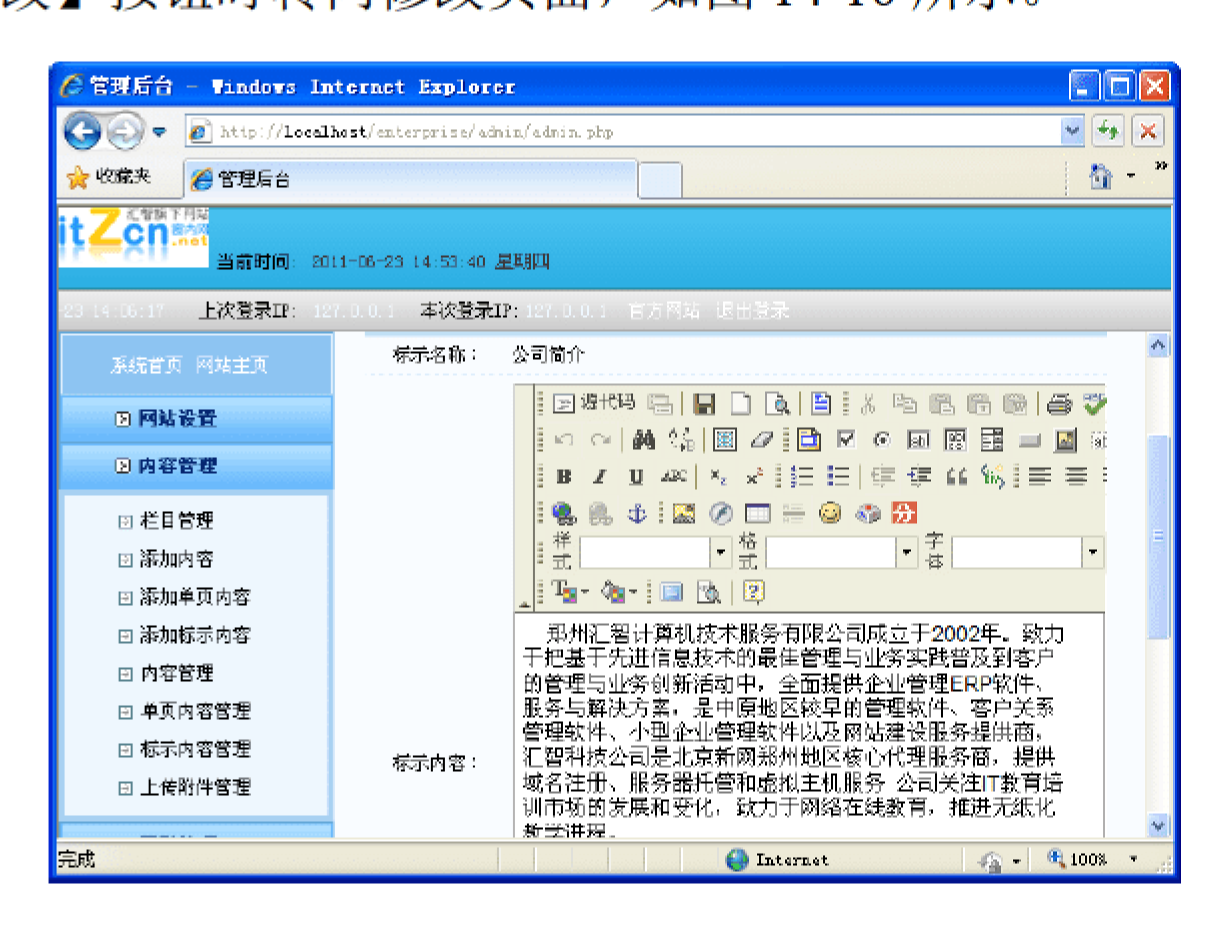


图 14-16　修改标示内容

通过以上运行结果可以看到标示内容，但不可以进行修改，在修改内容之前需要先将该标示信息之前的内容显示在修改页面中，方便用户的修改和参考，详细代码如下所示：

```
function edit_content(){
    if(!check_purview('content_edit')){msg('<span style="color:red">操作失败，你的权限不足!</span>');}
    global $id;
    if(empty($id)){
        msg('<span style="color:red">不存在相关内容</span>');
    }
    $arr=$GLOBALS['mysql']->fetch_asc("select*from ".DB_PRE."block where id={$id}");
    include('template/admin_content_tag_edit.html');
}
```

在该方法中，同样首先判断用户是否有权限修改标示内容信息，如果有，则获取修改标示内容的 id，通过查询将标示内容保存至变量中，最后转向模板页面呈现在用户面前。

呈现时是用户对信息内容的修改，但是用户只能修改信息的内容，而不可以修改标题，详细代码如下所示：

```
function save_edit(){
    if(!check_purview('content_edit')){msg('<span style="color:red">操作失败，
        你的权限不足!</span>');}
    global $id,$tag,$content;
    if(file_exists(DATA_PATH."sys_info.php")){include(DATA_PATH."sys_info.ph
        p");}
   $tag=empty($_sys['web_content_title'])?cn_substr($tag,60):cn_substr($tag,
        intval($_sys['web_content_title']));
    if(empty($id)){msg("参数传递错误,请重新操作");}
```

```
    $sql="update ".DB_PRE."block set content='{$content}' where id={$id}";
    $GLOBALS['mysql']->query($sql);
    msg("内容修改完成",'?action=content_list');
}
```

在上述代码中，判断是否有权限进行此操作，通过认证定义全局变量，用来保存修改标示信息内容、标题和 id 等信息，接下来则是对一些参数信息的判断，然后是拼接需要执行的 SQL 语句和修改内容，最后执行 query()方法执行修改操作，修改成功提示用户。

修改功能的实现即是对标示内容的删除操作功能的实现。当用户发现有部分内容不需要，可以使用该功能进行删除操作，详细代码如下所示：

```
function del(){
    if(!check_purview('content_del')){msg('<span style="color:red">操作失败,
        你的权限不足!</span>');}
    global $id;
    if(empty($id)){msg('<span style="color:red">参数错误,请重新操作</span>');}
    $GLOBALS['mysql']->query("delete from ".DB_PRE."block where id={$id}");
    msg('成功删除','admin_content_tag.php?action=content_list');
}
```

该功能实现非常简单，首先判断是否有权利进行删除操作，接下来是定义 id 全局变量来保存进行删除的标示内容 id，然后判断参数是否为空，最后拼接 SQL 语句执行删除操作，提示用户删除成功，转向列表页面。

14.7.3 上传附件管理

在上传文件管理模块中，有以下几个功能，获取当前文件下的所有文件列表、创建文件目录、上传文件、删除和修改文件操作，运行结果如图 14-17 所示。



图 14-17 上传文件管理列表

首先讲解将该目录下的目录和文件以列表呈现出来，代码如下所示：

```
function file_list(){
```

```
    if(!check_purview('file_manage')){msg('<span style="color:red">操作失败,
        你的权限不足!</span>');}
    global $path,$step,$file;
    $file_from=$file;
    $path=empty($path)?'upload':$path;
    $file_hand=@opendir(CMS_PATH.$path);
    if(!$file_hand){
        err("目录操作失败,请检查 upload 是否有足够的权限操作");
    }
    include('template/admin_file.html');
}
```

在上述代码中，同样首先判断该用户是否有对 upload 文件的操作权限，接下来则是判断路径是否为空，如果为空则将路径定义为 upload，否则定义为指定的路径，使用 opendir()方法打开当前路径，转向列表显示模板。

在模板页面中，使用 while 循环将该目录下的文件目录循环遍历出来，代码如下所示：

```
<?php
    readdir($file_hand);
    readdir($file_hand);
    while(false!=($file=readdir($file_hand))){
    $arr=get_dir_file($file,$path.'/');
    ?>
        <tr><td style="width:20%;text-align:center" align="center"><?php echo
$arr['path']?></td><td style="width:30%;text-align:center" align="center"><?php
echo empty($arr['size'])?' ':$arr['size'];?></td><td style="width:30%;
text-align:center"> <?php echo $arr['mtime']?></td><td style="width:20%;text-
align:center" align="center"><a href="javascript:if(confirm('确定要删除么,删除
后不可恢复!')){location.href='?action=del&file=<?php echo $path.'/'.$file;?>';}">
删 除 </a>|<a  href="?step=copy&file=<?php  echo  $file;?>&path=<?php  echo
$path;?>">复制</a></td></tr>
        <?php
        }
?>
```

在上述代码中使用循环遍历 readdir()中的文件信息，在循环中使用 get_dir_file()方法将文件信息保存至 arr 数组中，显示在列表中。

实现了文件列表的显示，接下来则是对文件的一些操作，创建目录是该功能下不可缺少的一项操作，如何实现该功能呢？代码如下所示：

```
function creat_dir(){
    if(!check_purview('file_manage')){msg('<span style="color:red">操作失败,
        你的权限不足!</span>');}
    global $path,$dir_name;
    if(!ereg('^[0-9a-z][0-9a-z]*[0-9a-z]$',$dir_name)){
        msg('<span style="color:red">名称只能是数字和字母</span>');
    }
    mkdir(CMS_PATH.$path.'/'.$dir_name,'0700');
```

```
    msg("目录【{$dir_name}】创建成功");
}
```

在上述代码中，判断当前登录用户是否有权限进行此操作，接下来是定义创建目录的名称和路径，使用正则表达式判断输入的文件目录名称是否合法，如果不合法则提示用户，否则执行 mkdir()方法进行创建文件夹目录，该方法有两个参数，第一个是路径，第二个是权限，创建目录成功提示用户。

目录创建好了，接下来是在目录下上传文件，将本地文件上传至服务器中，从而进行管理和应用，代码如下所示：

```
function up_load(){
    if(file_exists(DATA_PATH.'sys_info.php')){
        include(DATA_PATH.'sys_info.php');
    }
    if(!$_sys['web_upload_image']||!$_sys['web_upload_file']){
        msg('<span style="color:red">请先配置系统的上传文件</span>',
            'admin_sys.php');
    }

    $type=$_sys['web_upload_image'].'|'.$_sys['web_upload_file'];
    $type=explode('|',$type);
    global $path,$up,$name;
    $file=$path.'/'.up_file($_FILES['up'],$_sys['upload_size'],$type,$path,$
        name);
    msg("【{$file}】文件上传成功");
}
```

在上述代码中，首先判断用户选择上传文件的目录是否存在，如果存在则将文件上传至当前用户选择的目录下，如果上传路径不存在则提示用户请先配置系统的上传文件，使用 explode()方法来获取上传文件的类型，然后将上传文件的路径拼接起来，使用 up_file()方法进行上传，该方法有几个参数进行补充，上传成功则提示用户文件上传成功。

文件上传成功后，接下来是对文件的删除，有的时候难免会上传错误文件或需要对垃圾文件进行删除，因此对文件的操作也添加上了对文件删除的功能，实现该功能代码如下所示：

```
function del(){
    if(!check_purview('file_manage')){msg('<span style="color:red">操作失败,
        你的权限不足!</span>');}
    global $file;
    $sys_dir=array('upload/img','upload/fck','upload/file');
    if(in_array($file,$sys_dir)){msg('<span style="color:red">不能删除系统文件
        夹,只能删除内部内容</span>');}
    delete_dir($file);
    err('文件成功删除');
}
```

在代码中可以看到，要实现删除功能，首先需要判断当前用户是否有权利进行删除操作，然后定义删除文件路径的全局变量，然后判断当前文件是否是系统文件，如果不是则执行 delete_dir()方法进行删除，该方法有一个参数为删除文件的路径，删除成功则提示用户。

14.8 用 户 管 理

用户管理是针对站内的所有用户包括管理员和会员等用户进行管理的一个模块，在该模块中有 4 项功能，分别为管理员管理、管理员分组、会员管理和会员分组等，本节将使用两部分功能进行讲解。

14.8.1 会员分组

这里讲到的会员是前台页面进行登录的会员用户。为什么需要添加会员分组呢？原因是为方便管理人员对会员用户的管理和分配，使用会员分组，可以让管理人员很方便而且更加快速地对这些会员进行操作。

在这里首先讲解添加会员分组功能，在该功能中添加会员分组是非常简单的一项功能，在添加会员分组窗口中有 3 项内容，会员组名称、会员组描述和是否禁用等，运行结果如图 14-18 所示。

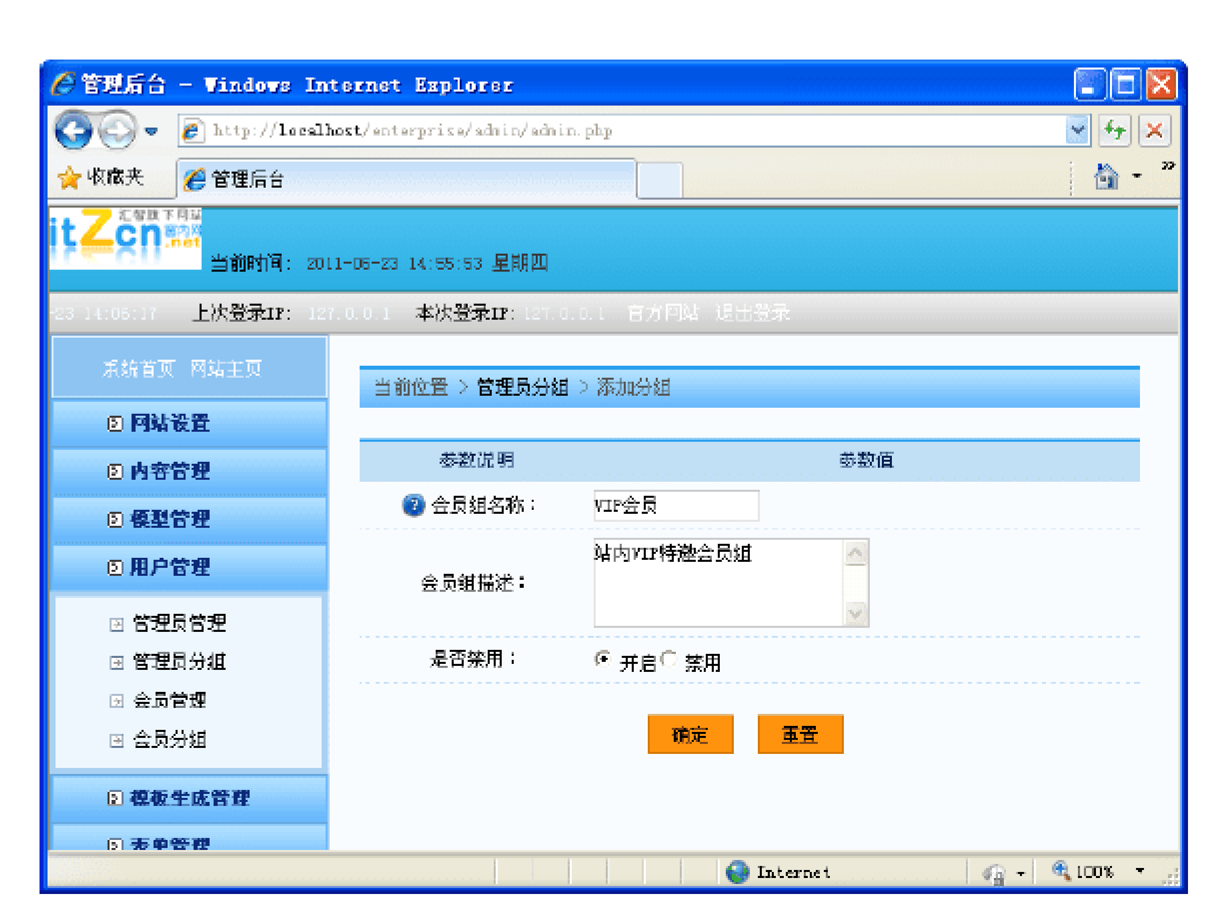


图 14-18　添加会员组

会员组名称是不可以重复的，如果已经存在该分组是无法重复添加同名的分组的。会员描述是对会员分组里面用户的一些简单描述。是否禁用，是指该分组是否进行使用，如果禁用则该类分组是无法使用的。

```
function save_membergroup(){
    if(!check_purview('user_group')){msg('<span style="color:red">操作失败,你
        的权限不足!</span>');}
    global $member_group_name,$member_group_info,$is_disable;
    if(empty($member_group_name)){
        msg('<span style="color:red">组名不能为空</span>');
    }
    $table=DB_PRE."member_group";
   $sql="insert into {$table} (member_group_name,member_group_info,is_disable)
```

```
values ('{$member_group_name}','{$member_group_info}',{$is_disable})";
    $GLOBALS['mysql']->query($sql);
    $GLOBALS['cache']->cache_member_group();
    msg('会员组"'.$member_group_name.'"添加成功','?action=member_group');
}
```

在上述代码中，同样判断用户是否有操作权限，接下来是定义分组的名称、分组描述和是否启用全局变量。判断输入组名称是否为空，如果不为空则拼接 SQL 语句调用 query()方法执行插入操作。

添加分组已经完成了，接下来是对分组的一些删除和修改的操作，首先进行修改会员组的操作，在修改之前需将当前需要进行修改的组信息显示在修改页面中，然后对此信息进行修改，代码如下所示：

```
function save_membergroup_edit(){
    if(!check_purview('user_group')){msg('<span style="color:red">操作失败,你
的权限不足!</span>');}
    global $id,$member_group_name,$member_group_info,$is_disable;
    if(empty($id)){
        msg('<span style="color:red">参数传递错误,请重新操作</span>');
    }
    if(empty($member_group_name)){
        msg('<span style="color:red">组名不能为空</span>');
    }
    $table=DB_PRE."member_group";
    $sql="update  {$table}  set  member_group_name='{$member_group_name}',
member_group_info='{$member_group_info}',is_disable={$is_disable}      where
id={$id}";
    $GLOBALS['mysql']->query($sql);
    $GLOBALS['cache']->cache_member_group();
    msg('会员组"'.$member_group_name.'"修改成功','admin_member.php?action=
        member_group');
}
```

上述代码是用来修改会员分组信息的代码，判断当前登录用户是否有权限进行修改操作，首先获取判断当前会员组 id 是否存在，如果存在则判断用户输入组名称是否为空，如果通过验证，则将 SQL 语句拼接执行 query()方法进行修改操作。

接下来是会员分组的删除功能，该功能是对该会员组进行删除，删除后该组中的会员用户将不会被删除，详细代码如下所示：

```
function membergroup_del(){
    global $id;
    $id=isset($id)?$id:'';
    if(empty($id)){msg('<span  style="color:red">参数发生错误，请重新操作
</span>','?action=member_group');}
    $GLOBALS['mysql']->query('delete from '.DB_PRE.'member_group where id='.$id);
    $GLOBALS['cache']->cache_member_group();
    msg('会员组删除成功');
}
```

代码中，首先获取需要删除分组的 id，判断 id 是否为空，如果为空则提示用户，否则拼接 SQL 语句，执行删除操作。

14.8.2　会员管理

会员管理模块主要是针对站内的会员用户进行管理，管理员不仅可以通过前台来注册会员，而且还可以通过后台的添加功能添加会员，一般在前台注册的会员用户默认都在注册会员分组中，而通过后台添加的用户则可以对其分组进行选择，同样还可以针对会员进行修改和删除操作，运行结果如图 14-19 所示。



图 14-19　会员列表

这里需要说明的是通过后台添加用户需要提供用户的登录用户名、登录用户密码、用户昵称、用户姓名、电子邮箱和联系电话等信息，运行结果如图 14-20 所示。

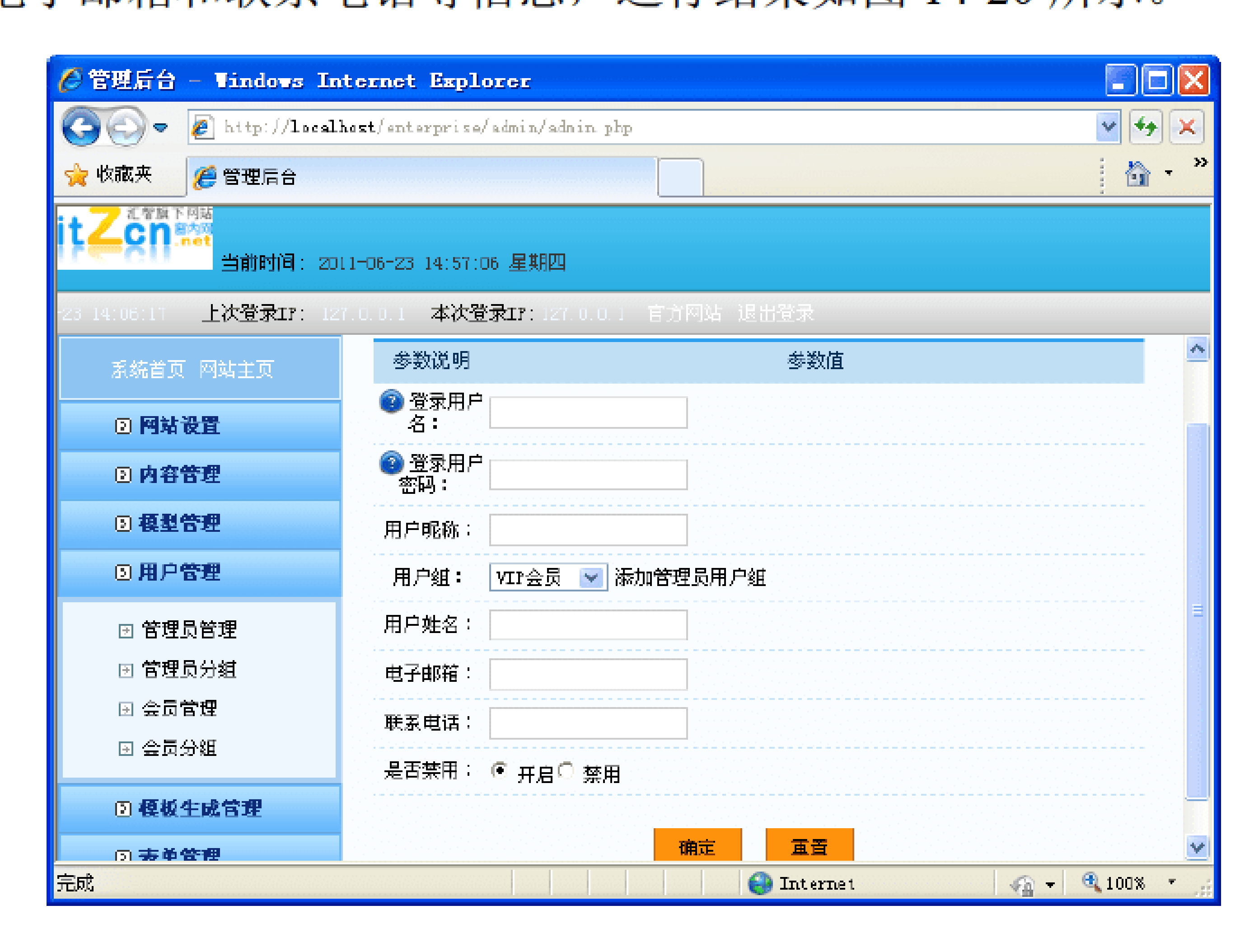


图 14-20　添加会员

在提交添加表单后，在后台将会对提交信息进行验证，验证提交内容为空和是否合法，同时判断当前添加用户是否已经被注册，详细代码如下所示：

```
function save_member(){
    if(!check_purview('user_manage')){msg('<span style="color:red">操作失败，
        你的权限不足!</span>');}
    global
$member_name,$member_password,$member_nich,$member_purview,$member_user,$mem
ber_mail,$member_tel,$is_disable;
    if(strlen($member_user)<3){
        msg('<span style="color:red">用户名长度必须大于 3</span>');
    }
    $is_member=$GLOBALS['mysql']->fetch_rows('select id from '.DB_PRE."member
        where member_user='".$member_user."'");
    if($is_member){msg('<span style="color:red">登录用户名【'.$member_user.'】
        已经存在，请更换</span>');}
    if(strlen($member_password)<3){
        msg('<span style="color:red">用户密码长度必须大于 3</span>');
    }
    if(!check_str($member_user,'/^[a-z0-9][a-z0-9]*$/')){
        msg('<span style="color:red">用户名必须由字母和数字组成</span>');
    }
    if(!check_str($member_password,'/^[a-z0-9!@#$%][a-z0-9!@#$%]*$/')){
        msg('<span style="color:red">用户密码包含其他不允许的字符</span>');
    }

    if(empty($member_nich)){
        msg('<span style="color:red">用户昵称不能为空</span>');
    }
    if(empty($member_purview)){
        msg('<span style="color:red">请选择用户组</span>');
    }
    if(!check_str($member_mail,'/^[0-9a-z]+@(([0-9a-z]+)[.])+[a-z]{2,3}$/')){
        msg('<span style="color:red">邮箱不正确</span>');
    }
    $is_mail=$GLOBALS['mysql']->fetch_rows('select id from '.DB_PRE."member
        where member_mail='".$member_mail."'");
    if($is_mail){msg('<span style="color:red">已经存在【'.$member_mail.'】邮箱，
        请更换</span>');}
    $table=DB_PRE."member";
    $member_password=md5($member_password);
    $sql="insert into {$table} (member_name,member_password,member_nich,member_purview,
member_user,member_mail,member_tel,is_disable,member_addtime) values ('{$member_name}',
'{$member_password}','{$member_nich}',$member_purview,'{$member_user}','{$m
ember_mail}','{$member_tel}',{$is_disable},".time().")";
    $GLOBALS['mysql']->query($sql);
    msg('会员"'.$member_name.'"添加成功','admin_member.php');
}
```

在代码中，首先判断输入用户名是否为空，长度是否合法，接下来查询数据库中该用户名是否被占用，如果没有被占用则使用正则表达式对用户密码、用户昵称、用户组和邮箱等信息进行验证，在对邮箱验证时需要在数据库内查询该邮箱是否被占用，如果没有占用则使用 md5()方法将用户密码进行 md5 加密，然后拼接至 SQL 语句中，调用 query()方法进行插入操作。

14.9　咨 询 管 理

当用户登录会员管理平台后，可以使用会员咨询模块功能向管理员咨询问题，此时后台管理人员将会在后台咨询管理模块查看会员咨询问题的详细信息，并且还可以对咨询内容进行回复。凡是被回复过的信息状态将会改变为“已回复”，后面显示的是回复时间，没有回复的信息状态是红色字体，“未回复”后面将没有回复时间，那么管理员将可以明确看到没有回复的咨询信息，该功能运行结果如图 14-21 所示。



图 14-21　咨询管理

14.9.1　回复咨询

管理员可以在咨询问题信息后端选择【回复查看】进入咨询问题回复页面，对会员用户咨询回复信息进行回复操作。在咨询回复页面，管理员只需填写回复内容即可，运行结果如图 14-22 所示。

在回复时，当回复内容为空时提交是无效的，在回复咨询列表状态仍是“未回复”，因此管理员需要输入回复内容对会员用户咨询的问题进行答复，提交回复信息后系统将会在该信息中添加数据，详细代码如下所示：

```
function save_reply(){
if(!check_purview('user_ask')){msg('<span style="color:red">操作失败,你的权限不
    足!</span>');}
global $id,$reply,$replytime;
if(empty($id)){msg("<span style=\"color:red\">参数错误</span>","?");}
$replytime=empty($replytime)?time():$replytime;
```

```
$sql="update  ".DB_PRE."ask  set  reply='{$reply}',replytime='{$replytime}'
where id={$id}";
$GLOBALS['mysql']->query($sql);
msg("咨询回复成功","?");
}
```



图 14-22　咨询回复

在上述代码中，首先判断当前登录管理员是否有权限进行此操作，通过验证后，判断接受的回复咨询问题 id 参数是否为空，然后判断回复时间是否为空，并赋予相应的值，拼接 SQL 语句调用 query()方法执行 update 修改操作，将回复信息保存至数据库中。

14.9.2　删除咨询

删除咨询问题也是必不可少的一项功能，有时可能用户咨询问题已经通过了回复，并且非常满意，为了节省空间和便于查询可以将以前咨询过的老问题进行删除，详细代码如下所示：

```
function del(){
if(!check_purview('user_ask')){msg('<span style="color:red">操作失败,你的权限不
    足!</span>');}
global $id;
$GLOBALS['mysql']->query("delete from ".DB_PRE."ask where id={$id}");
msg("咨询删除完成","?");
}
```

实现删除功能非常简单，只需要获取需要删除信息 id，将 id 信息拼接至 delete，删除 SQL 语句中调用 query()方法即可执行删除操作。

14.10　友 情 链 接

友情链接功能的主要作用是对网站与网站之间的相互宣传，可以加强用户对该网站之外相关的网站进行了解，因此在企业宣传网中也会插入该模块，该模块有对友情链接的添加、修改和删除等功能。图 14-23 为前台显示的友情链接。



图 14-23　友情链接

14.10.1　添加友情链接

在添加友情链接之前同样需要对用户提交表单进行验证，判断用户输入内容是否为空或者是否合法，需要插入的信息有网站网址、网站名称、网站 logo、显示顺序、网站说明、站长邮箱，运行结果如图 14-24 所示。

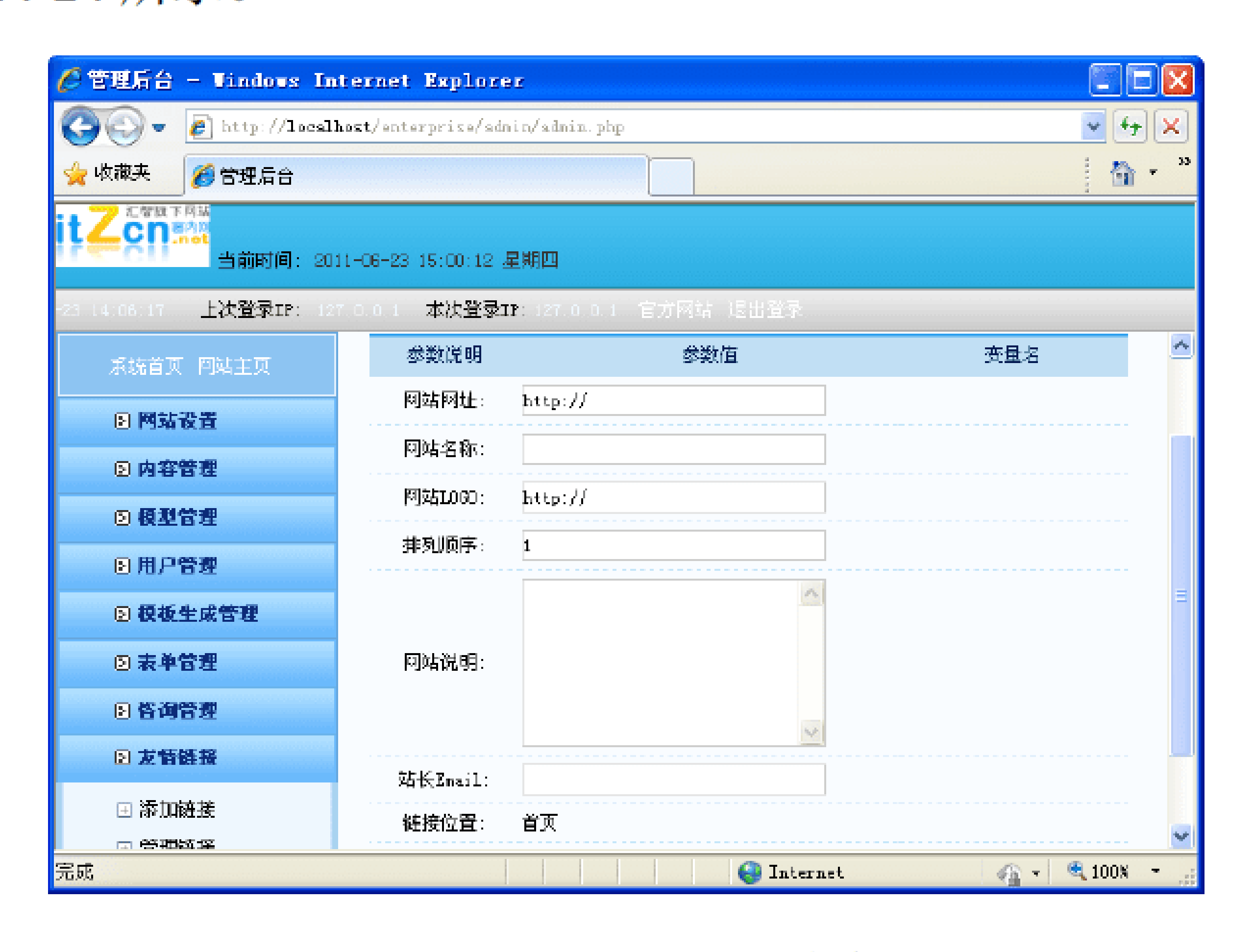


图 14-24　添加友情链接

在上述一系列的表单验证后，将用户提交的用户信息转向后台进行插入，在插入的过程中

还需要进行一些基本的验证方可执行插入方法，详细代码如下所示：

```
function save_add(){
if(!check_purview('link_add')){msg('<span style="color:red">操作失败,你的权限不足!</span>');}
global
$lang,$link_url,$link_name,$link_logo,$link_order,$link_info,$link_mail;
if(empty($link_url)){msg("<span style=\"color:red\">【网站网址】不能为空</span>");}
if(empty($link_name)){msg("<span style=\"color:red\">【网站名称】不能为空</span>");}
$link_order=empty($link_order)?1:$link_order;
$sql="insert into ".DB_PRE."link (link_url,link_name,link_logo,link_order,
link_info,link_mail,lang,addtime) values
('{$link_url}','{$link_name}','{$link_logo}',{$link_order},'{$link_info}','{
$link_mail}','{$lang}','".time()."')";
$GLOBALS['mysql']->query($sql);
msg("【{$link_name}】网站链接添加成功","?");
}
```

在上述代码中，定义常用的全局变量判断用户提交网站地址和网站名称时候为空，如果通过验证，则将拼接至 SQL 语句中，调用 query()方法，传入 SQL 语句执行 insert 操作。

14.10.2　编辑友情链接

编辑友情链接也是一项必不可少的功能，有时可能由于各种原因导致添加错误，没必要将信息删除后再添加，因此就添加了友情链接编辑的功能，首先将编写的友情链接基本信息展示在编辑页面中，用户可以通过对基本信息内容进行修改编写，编写后进行保存即可，同样，在编辑时需要对用户编写的内容进行基本的验证，运行结果如图 14-25 所示。

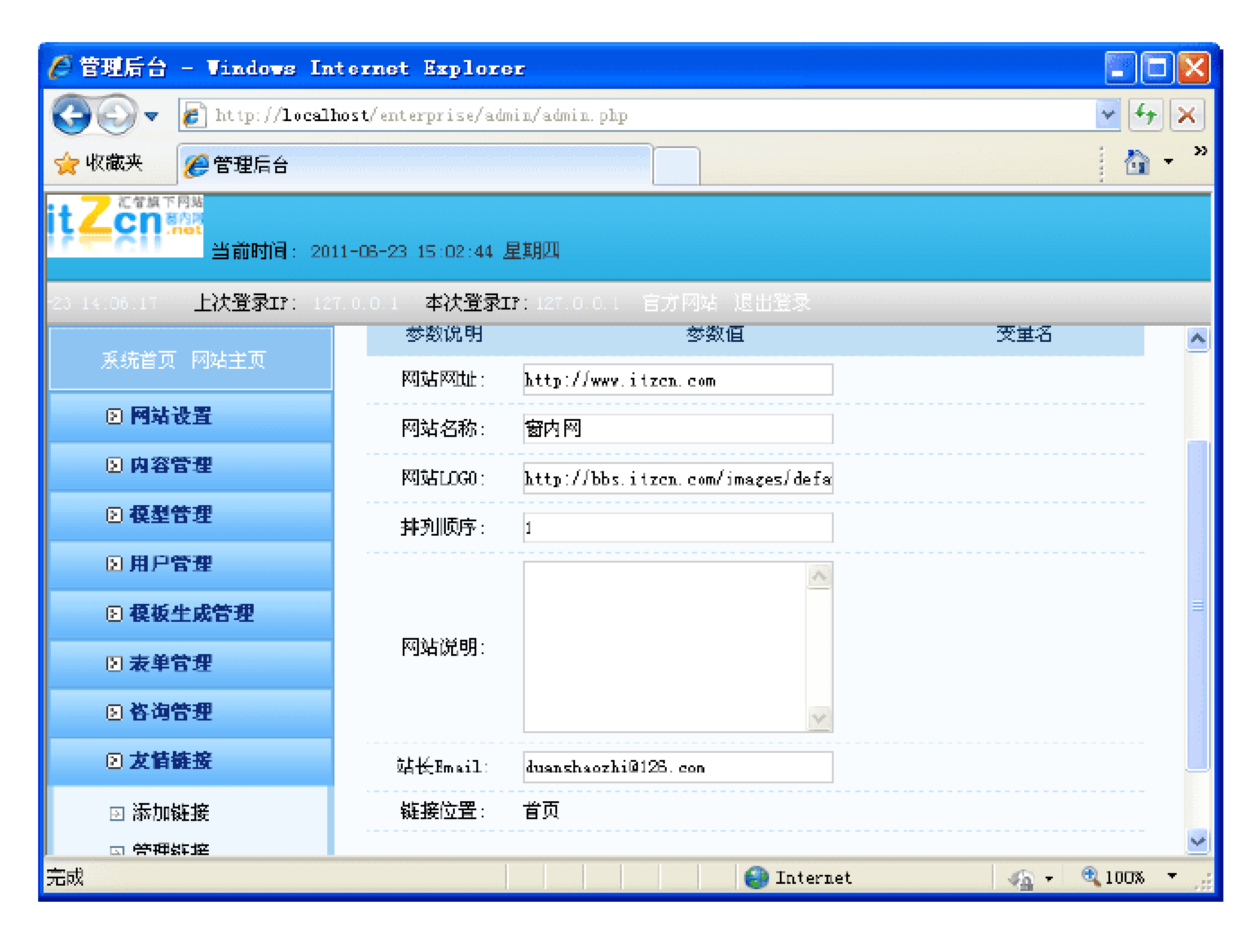


图 14-25　编辑友情链接

该功能的实现和添加友情链接非常相似，唯一不同的是最后执行的方法，一个是执行插入的 SQL 语句，另一个执行修改的 SQL 语句，详细代码如下所示：

```
function save_edit(){
    if(!check_purview('link_edit')){msg('<span style="color:red">操作失败,你的
        权限不足!</span>');}
    global $lang,$link_url,$link_name,$link_logo,$link_order,$link_info,
        $link_mail,$id;
    if(empty($link_url)){msg("<span style=\"color:red\">【网站网址】不能为空
        </span>");}
    if(empty($link_name)){msg("<span style=\"color:red\">【网站名称】不能为空
        </span>");}
    $link_order=empty($link_order)?1:$link_order;
    $link_info=empty($link_info)?'':cn_substr($link_info,255);
   $sql="update ".DB_PRE."link set link_url='{$link_url}',link_name='{$link_name}',
link_logo='{$link_logo}',link_order={$link_order},link_info='{$link_info}',
link_mail='{$link_mail}' where id={$id} and lang='{$lang}'";
    $GLOBALS['mysql']->query($sql);
    msg("【{$link_name}】网站链接修改成功",'?action=link_list');
}
```

首先判断当前登录用户是否有权限进行此操作，然后定义常量。判断用户提交的网站地址和网站名称是否为空，然后判断 link_order 是否为空，如果为空则设置为 1，然后判断 link_info 是否为空。最后拼接 SQL 语句执行 query()方法执行 update 修改操作。

14.10.3　删除友情链接

为了更方便地操作前台友情链接的显示，避免错误或者多余数据的冗留，因此就设定了该功能，详细代码如下所示：

```
function del(){
    if(!check_purview('link_del')){msg('<span style="color:red">操作失败,你的
权限不足!</span>');}
    global $lang,$id;
    if(empty($id)||empty($lang)){msg("<span style=\"color:red\">参数发生错误,
请重新操作</span>");}
    $GLOBALS['mysql']->query("delete from ".DB_PRE."link where id={$id} and
lang='{$lang}'");
    msg("成功删除链接",'?');
}
```

同样，在执行删除操作之前需要判断当前登录用户是否有权限进行此操作，通过验证之后获取需要删除友情链接的 id，判断 id 参数是否合法，然后指定 query()方法，将删除 SQL 拼接后参入该方法中执行 delete 删除操作。

14.11 前台首页

在企业宣传网系统中，最主要的部分是前台首页的展示功能。后台做的功能再好也是为了前台数据的显示而做的铺垫，为前台提供显示的详细数据信息。在前面已经讲解了内容管理和用户管理等功能，在这里我们将会用到这些功能添加的数据信息，图 14-26 所示是前台首页运行结果图。



图 14-26　首页

14.11.1 联系方式

在首页中显示联系方式的主要作用是让用户能够直观而且非常方便地发现宣传网站的联系方式和地址，能够增强网站的宣传力，因此在首页中将此信息显示在页面的右上方，运行结果如图 14-27 所示。

实现以上功能非常简单，只需要通过访问读取数据库，将数据库中信息读取出来绑定在该模块中即可，代码如下所示：

```
<div class="box_right"></div>
        <div class="box_title"><h2>{langs source='langs' name='contact'/}
</h2></div>
        <div class="box_in">
            <div class="contact" style="height:190px;">
            {block source=tag tpl="首页联系方式"/}
            </div>
        </div>
```

```
</div>
```



图 14-27　首页联系方式

将数据通过后台查询出来转向首页面，在首页面中使用以上代码，指定显示的标题信息使用 block 属性，接下来使用 langs 属性来绑定联系方式的信息内容，将 name 指定为后台传过来的 contact 属性值。

14.11.2　公司简介

公司简介模块的作用是介绍网站的后台公司的一些简单信息，能够让用户对宣传的网站有更多更深入的了解，激发客户对该公司产品的购买欲望，因此将该模块放在网站的中间部分，打开网站时可以直接看到该模块信息，运行结果如图 14-28 所示。



图 14-28　首页公司简介

该模块的实现和首页联系方式模块的实现非常相似，因此在这里就不过多啰嗦了，实现代码如下所示：

```
<div class="jianjie">
```

```
        <div class="box">
        <div class="box_left"></div>
        <div class="box_right"></div>
    <div class="box_title"><h2>
{langs source='langs' name='index_company'/}</h2></div>
    <div class="box_in">
        <div class="j_content">
        {block source=tag tpl="首页公司简介"/}
        </div>
    </div>
    </div><!--容器结束-->
</div><!--简介-->
```

同样在代码中，首先定义 div 容器，使用 block 属性设置模块标题，然后指定 langs 标记的 name 属性来设置显示公司简介的详细信息内容。

14.11.3 新闻中心

新闻中心是在首页中显示一些站内新闻信息，为了方便用户可以直观地了解站内的一些动态信息内容，当用户选择某条信息时转向新闻信息详细页面，运行结果如图 14-29 所示。

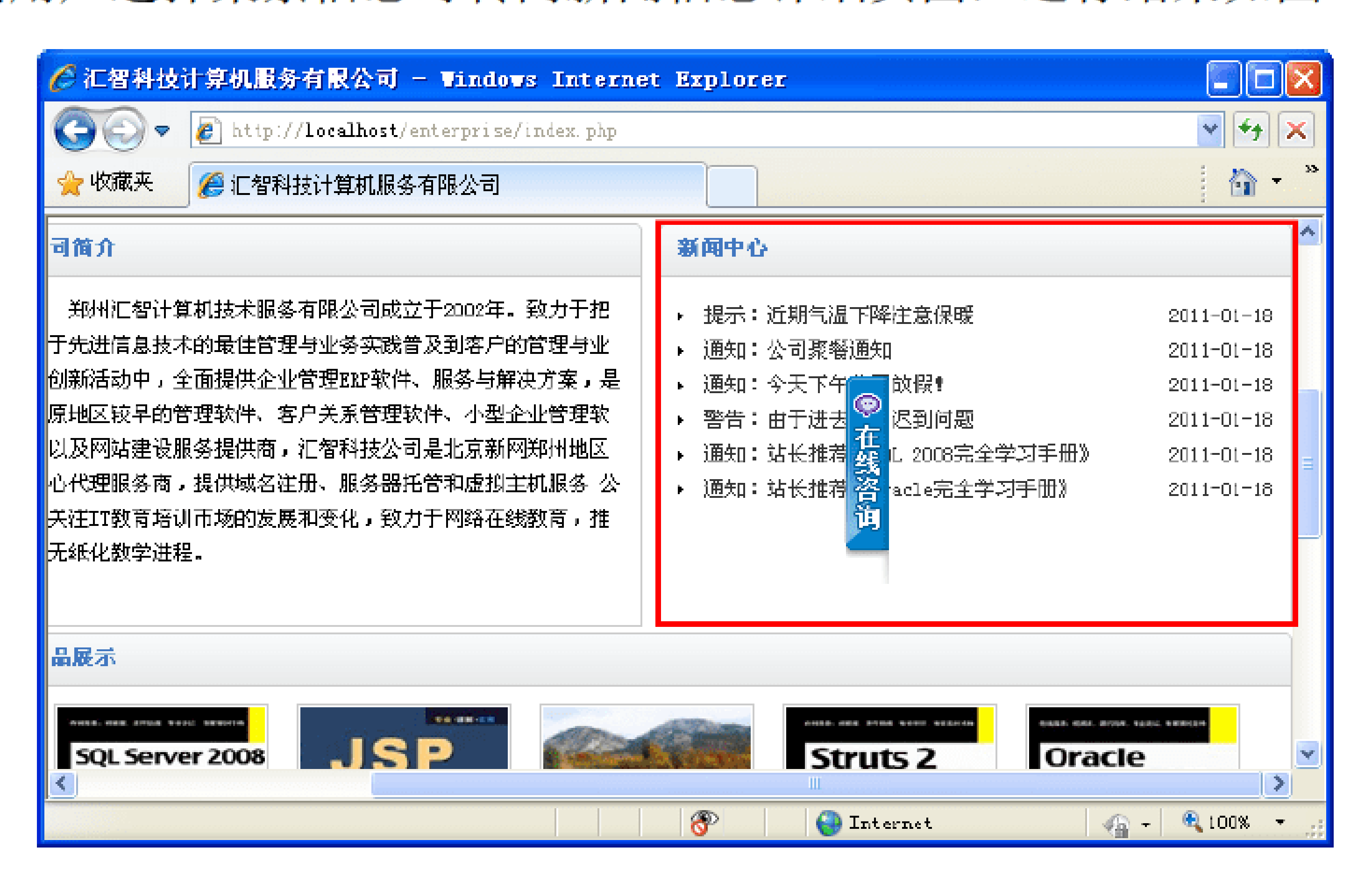


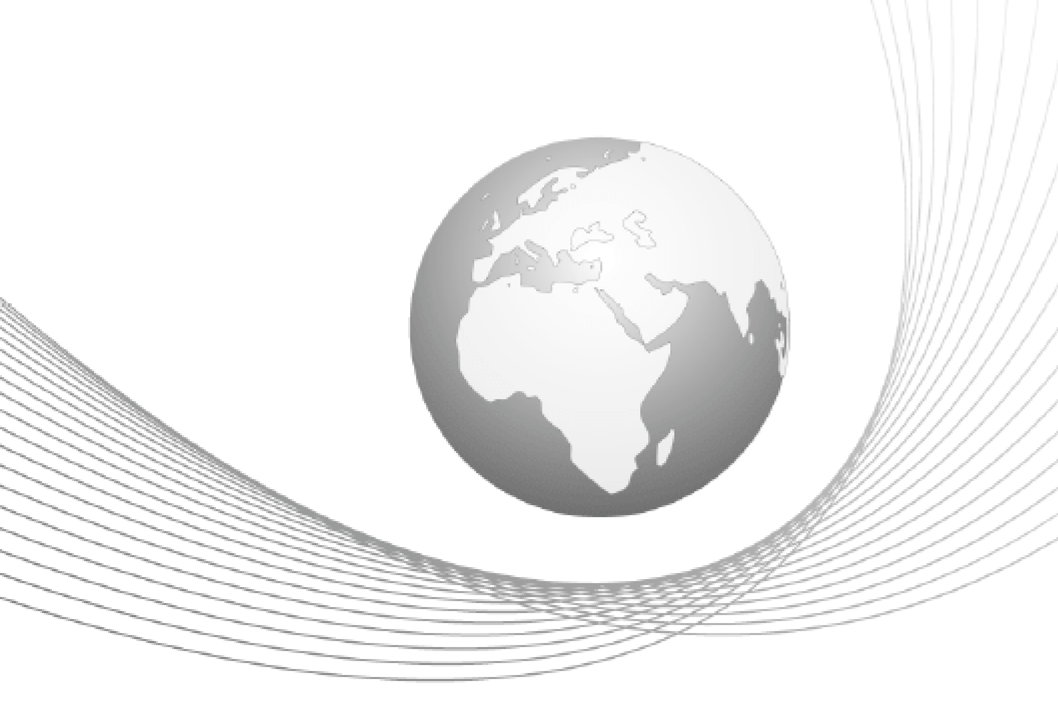
图 14-29 首页新闻中心

要想实现列表形式显示新闻信息，在这里就需要从数据库中读取数据集合，在前台页面中使用循环的形式将信息显示在该模块中，详细代码如下所示：

```
<div class="news_list">
        {loop source="article" titlelen="30" pics="yes" tpl="首页新闻中心图片显
            示" info="数量从 0 到 1，显示上传有缩略图的内容"}
            <div class="index_news_tj"><a href="{print $v['url']/}"><img
                height="90" width="120" src="{print $v['tbpic']/}"></a>
            <p class="title2"><a href="{print $v['url']/}" {print $v['target']/}
                title="{print $v['title']/}">{print $v['style_title']/}</a></p>
        <p class="info">{print cn_substr($v['info'],140)/}</p>
```

```
        </div>
        {/loop}
             <url>
             {loop source="article" titlelen="60" tpl="首页新闻中心"}
             <li><a title="{print $v['title']/}" href="{print $v['url']/}"
{print  $v['target']/}>{print  $v['style_title']/}</a>{print  date('Y-m-d',
$v['updatetime'])/}</li>
        {/loop}
            </ul>
            </div>
        </div>
</div>
```

在上述代码中使用 loop 来循环遍历 article 中的集合数据，然后以列表的形式显示在页面中，在页面中指定了新闻标题信息、新闻详细地址和新闻发布时间，然后使用 date 方法将时间格式化。



第 15 章　电子商务平台网

内容摘要：

电子商务平台网可以让顾客足不出户，直接在网上进行商品预定，并接收用户在线订单。该网站主要采用 PHP+MySQL 技术实现简单的电子商务平台功能，并分为前台与后台两大功能实现，前台主要实现了顾客登录与注册、商品的展示、购买商品、去收银台结账等功能；后台提供了管理员对商品的管理、用户的管理、订单的管理等几个功能。本章将详细介绍如何开发电子商务平台网。

学习目标：

- 整体掌握电子商务平台网。
- 掌握网站前台首页显示的相关功能。
- 掌握如何实现商品的展示。
- 掌握如何实现登录与注册。
- 掌握如何实现用户中心的相关功能。
- 掌握如何实现商品的购买和去收银台结账。
- 掌握网站后台商品管理的相关功能。
- 掌握网站后台用户管理的相关功能。
- 掌握网站后台订单管理的相关功能。
- 掌握网站后台信息管理的相关功能。

15.1 网站概述

电子商务(Electronic Commerce)是在因特网开放的网络环境下，基于浏览器/服务器应用方式，实现消费者的网上购物、商户之间的网上交易和在线电子支付的一种新型的商业运营模式。

随着互联网技术的飞速发展，互联网已经走进千家万户，然而中国各个行业的营销方式仍然传统落后，独立分散。网上商店为了突破时空限制，降低交易成本，节约客户订购、支付和配送的时间，方便客户购买，决定进入电子商务网的销售市场，建立一个网上销售系统，利用互联网在线支付平台进行交易，实现网络营销与传统营销双通道同时运行的新型营销模式。

网上商店采用现有的各种网络技术，构建商品多级查询、选择、订购的一个网上销售系统，为客户提供方便、快捷、安全的网上购物环境。

15.2 需求分析

网上购物作为新生的一种购物形式，渐渐地走入我们的日常生活。它的优势在于足不出户的消费者们凭着网络强大的搜索功能，不需要东奔西跑就能瞬间将需要的产品大致浏览，轻松做到货比三家，甚至三十家，轻松对比商品价格。而最大的缺点及潜在的问题就是商家诚信、支付安全性、所购物品的质量及退货换货问题。商家诚信通过信用评价可基本了解。付款方式选择第三方交易工具比较安全，例如支付宝、财付通、安付通、贝安等。质量问题双方按交易规定可以解决，只是过程会有点繁琐。如果可以做到货到付款，这些风险就会相对减低，但很多商家都不支持货到付款。

一般电子商务网站的用户购物流程如图 15-1 所示。

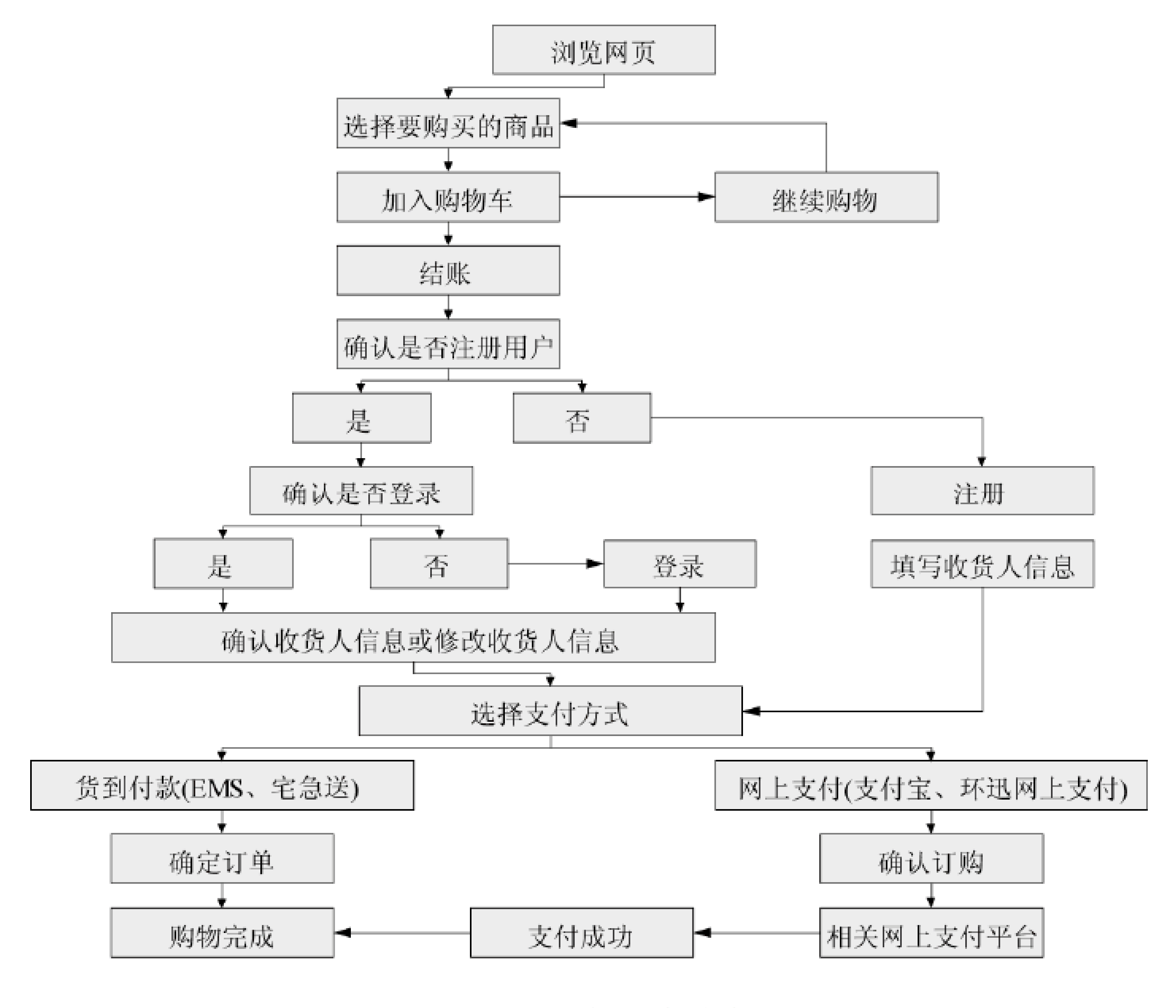


图 15-1　用户购物流程图

根据上述需求分析，将电子商务平台网前台的主要功能模块划分为：首页显示、商品展示、登录与注册、用户中心、商品购买、结账等。前台的详细功能分析如图 15-2 所示。

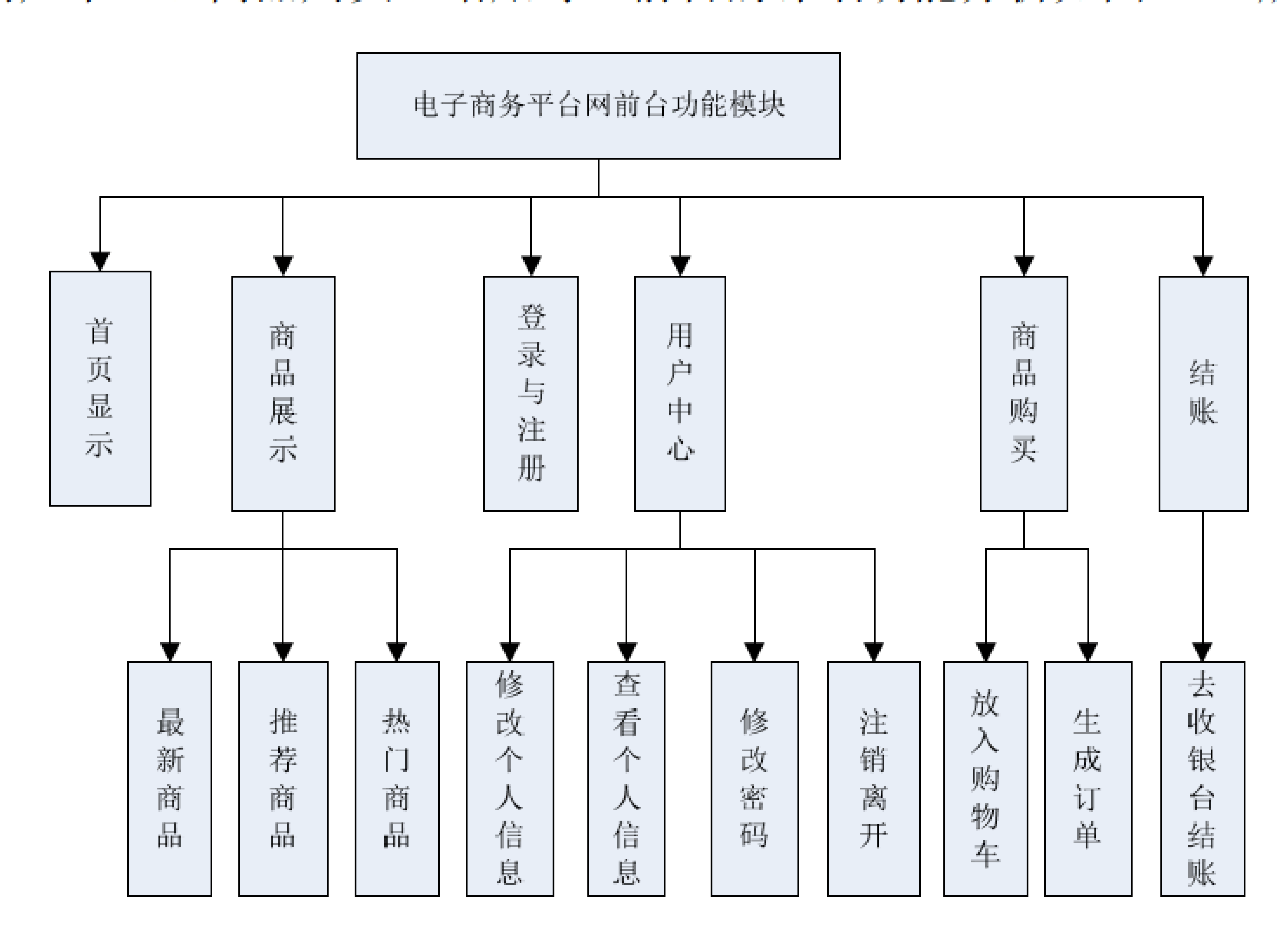


图 15-2　电子商务平台网前台功能模块

电子商务后台的主要功能包括商品管理、用户管理、订单管理、信息管理等。后台的详细功能分析如图 15-3 所示。

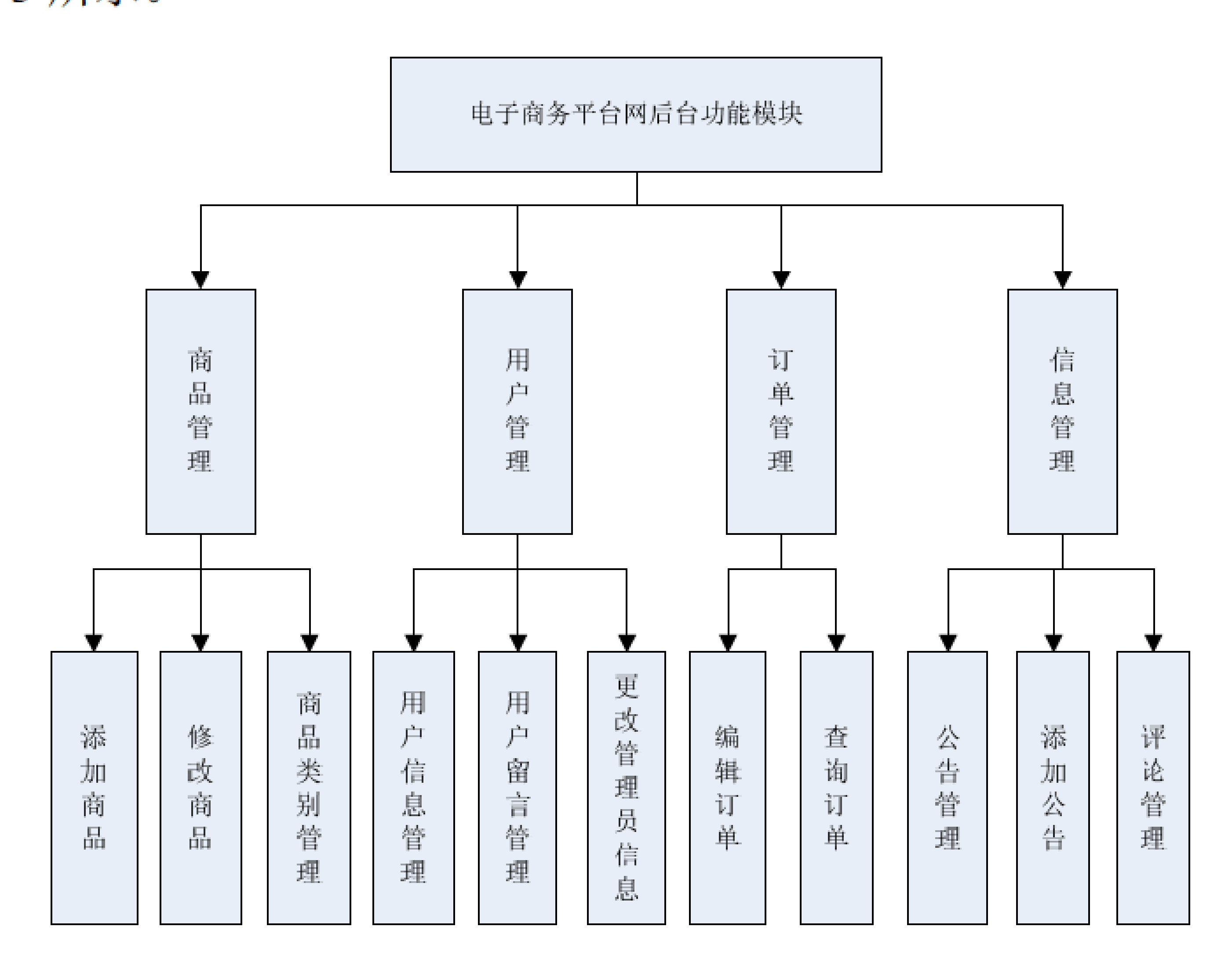


图 15-3　电子商务平台网后台功能模块

15.3　数据库设计

电子商务平台网使用了 MySQL 数据库，设计数据库名字为 db_shop，通过上述系统设计分析后，在数据库中设计出 9 张表，分别为管理员用户表、订单表、公告表、用户留言信息表、友情链接表、用户评论表、商品信息表、商品类型表、普通用户表。下面分别对每张表的详细

设计进行讲解。

1. 管理员用户表

管理员用户表用于保存管理员的信息，表中定义了 3 个字段，管理员编号(id)、管理员名称(name)和密码(pwd)。管理员用户表的详细设计如表 15-1 所示。

表 15-1 管理员用户表

字段名称	类 型	约 束	含 义
id	int	主键	管理员编号
name	varchar	非空	管理员名称
pwd	varchar	非空	管理员密码

2. 订单表

订单表用于保存用户购买商品生成的订单信息。该表中定义了 17 个字段，分别为订单 id 号、订单编号(dingdanhao)、购买商品的编号(spc)、购买商品的数量(slc)、收货人(shouhuoren)、性别(sex)、地址(dizhi)、邮编(youbian)、电话号码(tel)、电子邮件(email)、送货方法(shff)、支付方法(zfff)、用户留言(leaveword)等。订单表的详细设计如表 15-2 所示。

表 15-2 订单表

字段名称	类 型	约 束	含 义
id	int	主键	订单 id 号
dingdanhao	varchar	可空	订单编号
spc	varchar	可空	购买商品的编号
slc	varchar	可空	购买商品的数量
shouhuoren	varchar	可空	收货人
sex	varchar	可空	性别
dizhi	varchar	可空	地址
youbian	varchar	可空	邮编
tel	varchar	可空	电话
email	varchar	可空	电子邮件
shff	varchar	可空	送货方法
zfff	varchar	可空	支付方式
leaveword	mediumtext	可空	用户留言
time	varchar	可空	生成订单时间
xiadanren	varchar	可空	下单人
zt	varchar	可空	订单状态
total	varchar	可空	订单数量

3．公告表

公告表用于保存该网站商品的促销特价活动信息，该表中定义了 4 个字段，分别是公告 id 号(id)、标题(titile)、内容(content)和时间(time)。该表的详细设计如表 15-3 所示。

表 15-3　公告表

字段名称	类　型	约　束	含　义
id	int	主键	公告 id 号
title	varchar	可空	标题
content	text	可空	内容
time	varchar	可空	时间

4．用户留言信息表

用户留言信息表用于保存用户留言的内容，该表定义了 5 个字段，分别是用户留言 id 号(id)、用户编号(userid)、标题(titile)、内容(content)和时间(time)。该表的详细设计如表 15-4 所示。

表 15-4　用户留言信息表

字段名称	类　型	约　束	含　义
id	int	主键	用户留言 id 号
userid	int	外键	用户编号
title	varchar	可空	留言标题
content	text	可空	留言内容
time	varchar	可空	时间

5．友情链接表

友情链接表用于保存一些帮助用户更好地在该网站中进行购物的链接信息。该表中定义了 3 个字段，分别是编号(id)、链接名称(linkname)、链接 url(url)。该表的详细设计如表 15-5 所示。

表 15-5　友情链接表

字段名称	类　型	约　束	含　义
id	int	主键	友情链接 id 号
linkname	varchar	可空	链接名字
linkurl	varchar	可空	链接 url

6．用户评论表

用户评论表用于保存用户的评论信息，该表中定义了 6 个字段，分别是评论 id 号(id)、用户编号(userid)、商品编号(spid)、评论标题(titile)、评论内容(content)和时间(time)。该表的详细设计如表 15-6 所示。

表 15-6　用户评论表

字段名称	类　型	约　束	含　义
id	int	主键	评论 id 号
userid	int	外键	用户编号
spid	int	外键	商品编号
title	varchar	可空	评论标题
content	text	可空	评论内容
time	varchar	可空	时间

7．商品信息表

商品信息表用于保存该网站所有商品的信息，该表中定义了 14 个字段，分别是商品编号(id)、商品名称(mingcheng)、简介(jianjie)、添加时间(addtime)、等级(dengji)、型号(xinghao)、图片(tupian)、数量(shuliang)、浏览次数(cishu)、是否为推荐商品(tuijian)、商品类型(typeid)等等。该表的详细设计如表 15-7 所示。

表 15-7　商品信息表

字段名称	类　型	约　束	含　义
id	int	主键	商品编号
mingcheng	varchar	可空	商品名称
jianjie	mediumtext	可空	商品简介
addtime	varchar	可空	添加时间
dengji	varchar	可空	等级
xinghao	varchar	可空	型号
tupian	varchar	可空	图片
shuliang	int	可空	数量
cishu	int	可空	浏览次数
typeid	int	可空	商品类型
huiyuanjia	varchar	可空	会员价
shichangjia	varchar	可空	市场价
pinpai	varchar	可空	品牌
tuijian	int	可空	是否推荐

8．商品类型表

商品类型表用于保存商品类别信息，该表中定义了两个字段，分别是类型 id 号(id)和类型名称(typename)。该表的详细设计如表 15-8 所示。

表 15-8　商品类型表

字段名称	类　型	约　束	含　义
id	int	主键	类型 id 号
typename	varchar	可空	类型名称

9．普通用户表

普通用户表用于保存在该网站购物用户的信息，该表中定义了 15 个字段，分别是用户 id 号(id)、用户名称(name)、密码(pwd)、是否被冻结(dongjie)、电子邮件(email)、身份证号(sfzh)、电话(tel)等。该表的详细设计如表 15-9 所示。

表 15-9　普通用户表

字段名称	类　型	约　束	含　义
id	int	主键	用户 id 号
name	varchar	可空	用户名称
pwd	varchar	可空	密码
dongjie	int	可空	是否冻结
email	varchar	可空	电子邮件
sfzh	varchar	可空	身份证号
tel	varchar	可空	电话
qq	varchar	可空	QQ 号
tishi	varchar	可空	找回密码问题提示
huida	varchar	可空	问题答案
dizhi	varchar	可空	地址
youbian	varchar	可空	邮编
regtime	varchar	可空	注册时间
truename	varchar	可空	真实姓名
pwd1	varchar	可空	确认密码

15.4　连接数据库

不管后台还是前台都要与数据库进行交互，因此本节将讲解如何使用 PHP 脚本来连接数据库 db_shop。项目 shop/conn 文件夹下的 conn.php 文件用来完成连接数据库的操作。代码如下所示：

```
<?php
    //连接 MySql 数据库传入用户名与密码，得到一个连接数据库对象$conn
    $conn=mysql_connect("localhost","root","123456") or die("数据库服务器连接错
        误".mysql_error());
    //设置连接数据库的名字为 db_shop
    mysql_select_db("db_shop",$conn) or die("数据库访问错误".mysql_error());
    //设置数据库字符集编码为 gb2312
    mysql_query("set character set gb2312");
    //设置客户端字符集
    mysql_query("set names gb2312");
?>
```

15.5 前台模块实现

本网站前台主要是针对购买网站商品的用户设计的，没有注册本网站账号只能浏览网站商品，不能执行购买商品以及其他操作。网站前台主要提供了首页显示、商品展示、登录与注册、用户中心、商品购买、去收银台结账 6 个模块。本节将详细讲解这 6 个模块的实现。

15.5.1 首页显示

电子商务平台网首页主要是方便展示本网站的各种特色商品，例如最新商品、推荐商品、热门商品等。还有方便用户登录与注册，显示商场公告信息。用户登录之后还可以单击【用户中心】链接查看、修改用户信息等操作。下面先来看看该网站的首页执行效果，如图 15-4 所示。



图 15-4 电子商务平台网首页

怎么样？漂亮吧，实际上不仅漂亮而且一些常见的电子商务网站的各种首页功能，本网站首页也都提供了，很实用。比如说用户可以进行登录与注册，浏览一些网站推荐的商品信息，还有可以单击【购买】链接将商品放入购物车进行购买，单击【订单查询】链接查询用户订单信息等。

这些功能大多仅仅是在首页中提供了一个链接，具体实现将在下面进行详细讲解。首页详细代码可查看光盘文件本章目录中的 shop/index.php 文件。

15.5.2 商品展示

本网站商品分 3 种进行展示，分别是最新商品、推荐商品、热门商品。在登录网站首页即可看到这 3 种商品的部分信息，单击【最新商品】、【推荐商品】、【热门商品】链接可以查看这 3 种商品的全部信息。

1. 最新商品

最新商品是该网站推出的新产品，用户在查看最新商品信息的同时也可以将喜欢的商品加入购物车进行购买，实现代码如下所示：

```
<?php
    include("top.php"); //引入 top.php 文件，将 top.php 文件内容显示在该页面的顶部
?>
<table width="766" height="438" border="0" align="center" cellpadding="0"
cellspacing="0">
    <tr>
        <td width="209" height="438" valign="top" bgcolor="#F0F0F0">
            <?php include("left.php");?></td>
        <td width="557" align="center" valign="top" bgcolor="#FFFFFF">
            <table width="550" height="25" border="0" align="center"
                cellpadding="0" cellspacing="0">
                <tr>
                    <td height="50" background="images/new1.gif"> </td>
                </tr>
            </table>
            <table width="550" height="10" border="0" align="center"
                cellpadding="0" cellspacing="0">
                <tr>
                    <td background="images/line1.gif"></td>
                </tr>
            </table>
            <table width="550" height="70" border="0" align="center"
                cellpadding="0" cellspacing="0">
                <?php       //从数据库中查询最新商品信息
                    $sql=mysql_query("select * from tb_shangpin order by
                        addtime desc limit
                        0,4",$conn);
                    $info=mysql_fetch_array($sql);      //获取是否查询到最新商品
                    if($info==false){        //没有的的话显示没有最新商品
                        echo "本站暂无最新产品!";
                    }
                    else{
                        do{      //查询到商品的话，通过循环依次迭代取出商品进行显示
                ?>
                <tr>
                    <td width="89"  rowspan="6"><div align="center">
                    <?php
                        if($info[tupian]==""){
                            echo "暂无图片!";
                        }
                        else{ ?>
                <a href="lookinfo.php?id=<?php echo $info[id];?>"><img border="0"
                    src="<?php echo $info[tupian];?>" width="80" height="80"></a>
                <?php } ?>
            </div></td>
            <td width="93" height="20"><div align="center" style="color:
```

```
            #000000"> 商品名称：</div></td>
            <td colspan="5"><div align="left">
            <a href="lookinfo.php?id=<?php echo $info[id];?>"><?php echo
                $info[mingcheng];?></a>
            </div></td>
            </tr>
            <tr>
                <td width="93" height="20">
                    <div align="center" style="color: #000000">商品品牌：</div></td>
                <td width="101" height="20"><div align="left">
                    <?php echo $info[pinpai];?></div></td>
                    <td width="62"><div align="center" style="color: #000000">
                        商品型号：</div></td>
                    <td colspan="3"><div align="left">
                        <?php echo $info[xinghao];?></div></td>
            </tr>
            <tr>
                <td width="93" height="20"><div align="center" style="color: #000000">
                    商品简介：</div></td>
                <td height="20" colspan="5"><div align="left">
                    <?php echo $info[jianjie];?></div></td>
            </tr>
            <tr>
                <td height="20"><div align="center" style="color: #000000">
                    上市日期：</div></td>
                <td height="20"><div align="left"><?php echo $info[addtime];?>
                    </div></td>
                <td height="20"><div align="center" style="color: #000000">
                    剩余数量：</div></td>
                <td width="69" height="20"><div align="left">
                    <?php echo $info[shuliang];?></div></td>
                <td width="63"><div align="center" style="color: #000000">商
                    品等级：</div></td>
                <td width="73"><div align="left"><?php echo $info[dengji];?>
                    </div></td>
            </tr>
            <tr>
                <td height="20"><div align="center" style="color: #000000">
                    商场价：</div></td>
                <td height="20"><div align="left"><?php echo $info[shichangjia];?>
                    元 </div></td>
                <td height="20"><div align="center" style="color: #000000">
                    会员价：</div></td>
                <td height="20"><div align="left"><?php echo $info[huiyuanjia];?>
                    元</div></td>
                <td height="20"><div align="center" style="color: #000000">
                    折扣：</div></td>
                <td height="20"><div align="left">
                <?php echo (@ceil(($info[huiyuanjia]/$info[shichangjia])
                    *100))."%";?></div></td>
            </tr>
```

```
            <tr>
                <td height="20" colspan="6" width="461">
                    <div align="center">    
                    <a href="addgouwuche.php?id=<?php echo $info[id];?>">
                    <img src="images/goumai_btn.gif" width="60" height="18"
                    border="0" style=" cursor:hand"></a></div></td>
            </tr>
            <tr>
                <td height="10" colspan="7" background="images/line1.gif">
                    </td>
            </tr>
            <?php
                    }while($info=mysql_fetch_array($sql));
                }
            ?>
        </table></td>
    </tr>
</table>
<?php
    include("bottom.php");
    //引入 bottom.php 文件，将 bottom.php 文件内容显示在该页面的底部
?>
```

上述代码首先通过 include()函数将 top.php 文件引入到该页面中，这样在执行该页面时会将 top.php 文件内容加载显示到该页面的顶部，然后定义了一个表格用来显示从数据库中查询得到的最新商品的信息，如商品名称、品牌、图片、型号、会员价、市场价等。还提供了将商品加入购物车的图片链接，用户可以单击该链接进行购买。最后通过 include()函数将 bottom.php 文件引入到该页面中，bottom.php 文件的内容将显示在该页面的底部。

当用户单击【最新商品】链接时，可以在页面中看到该网站所有最新商品的信息，执行效果如图 15-5 所示。



图 15-5 最新商品信息

2．推荐商品

除了最新商品外，本网站还将向用户推荐一些价格优惠、质量有保证、性价比比较高的商品，下面就来看看如何实现查看推荐商品的信息，由于在页面中显示推荐商品的基本信息(商品名称、品牌、市场价等)的实现与最新商品的大致相同，因此这部分代码就不再给出。详细实现代码可查看随书光盘本章目录中的 shop/showtuijian.php。这里只针对如何查询推荐商品信息和实现分页显示进行详细讲解，代码如下所示：

```
<?php
    //发出 SQL 语句查询推荐商品信息
    $sql=mysql_query("select count(*) as total f rom tb_shangpin where tuijian=1
",$conn);
    $info=mysql_fetch_array($sql);      //获取查询结果
    $total=$info[total];                //取出查询推荐商品的记录数
    if($total==0)                       //记录数为 0，显示没有推荐商品
    {
        echo "本站暂无推荐产品!";
    }
    else
    {        //将分页显示推荐商品信息
?>
<table  width="550"  height="70"  border="0"  align="center"  cellpadding="0"
cellspacing="0">
<?php
    $pagesize=20;            //设置每页显示 20 个推荐商品的信息
    if ($total<=$pagesize){ //判断查询得到的商品记录数小于或等于每页显示的个数,
                            //将显示页数指定为 1
        $pagecount=1;
    }
    if(($total%$pagesize)!=0){  //判断查询得到的记录数除以每页显示的个数是否有余数
        $pagecount=intval($total/$pagesize)+1;
                                //有余数的话，取整加 1，得到要显示的页数
    }else{
        $pagecount=$total/$pagesize;    //没有余数的话，直接得到要显示的页数
    }
    if(($_GET[page])==""){  //获取 page 参数
        $page=1;
    }else{
        $page=intval($_GET[page]);
    }
    //按照分页要显示的数据，重新计算查询每页要显示的商品信息数据
    $sql1=mysql_query("select  *  from  tb_shangpin  where  tuijian=1  order  by
addtime
    desc limit ".($page-1)*$pagesize.",$pagesize ",$conn);
```

上述代码先从数据库中查询到推荐商品的信息，然后设置推荐商品进行分页显示，每页显

示 20 个，下面代码是如何在页面中显示分页工具条：

```
<?php
    echo $total;
?>
 件 每页显示 <?php echo $pagesize;?> 件 第 
<?php echo $page;?> 页/共 <?php echo $pagecount; ?> 页
<?php
    if($page>=2)
    {
?>
<a  href="showtuijian.php?page=1"  title=" 首 页 "><font  face="webdings">  9
</font></a>
<a  href="showtuijian.php?id=<?php  echo  $id;?>&page=<?php  echo  $page-1;?>"
title="前一页">
<font face="webdings"> 7 </font></a>
<?php
    }
    if($pagecount<=4){
        for($i=1;$i<=$pagecount;$i++){
?>
<a href="showtuijian.php?page=<?php echo $i;?>"><?php echo $i;?></a>
<?php
        }
    }else{
        for($i=1;$i<=4;$i++){
?>
<a href="showtuijian.php?page=<?php echo $i;?>"><?php echo $i;?></a>
<?php }?>
<a href="showtuijian.php?page=<?php echo $page-1;?>" title="后一页">
<font face="webdings"> 8 </font></a>
<a href="showtuijian.php?id=<?php echo $id;?>&page=<?php echo $pagecount;?>"
title="尾页">
<font face="webdings"> : </font></a>
<?php }?>
```

上述代码通过前面的计算将在页面中显示分页工具条，如首页、前一页、后一页和尾页。单击首页【推荐商品】链接可以看到推荐商品的全部信息，执行效果如图 15-6 所示。

3．热门商品

热门商品展示的是最近本网站卖的特别火的商品，由于它的实现与最新商品的展示基本相同，因此就不再给出代码进行讲解了，该模块实现的详细代码可查看随书光盘本章目录中的 shop/showhot.php 文件，执行效果如图 15-7 所示。



图 15-6 推荐商品



图 15-7 热门商品

15.5.3　登录与注册

任何一个电子商务网站都要实现登录与注册，本网站也不例外，用户只有在该网站进行注册并登录之后才能进行购物，否则，只能浏览商品，不能进行购买、评论留言等操作。下面就先来看看本网站是如何实现登录的吧，代码如下所示：

```
<form  name="form2"  method="post"  action="chkuser.php"  onSubmit="return
chkuserinput(this)">
    <tr>
        <td height="10" colspan="3"> </td>
    </tr>
    <tr>
        <td width="50" height="20"><div align="right">用户：</div></td>
        <td height="20" colspan="2">
            <div align="left"><input type="text" name="username" size="19"
                class="inputcss" style="background-color:#e8f4ff "
                onMouseOver="this.style.backgroundColor='#ffffff'"
                onMouseOut="this.style.backgroundColor='#e8f4ff'">
            </div></td>
    </tr>
    <tr>
        <td height="20"><div align="right">密码：</div></td>
        <td colspan="2"><div align="left">
            <input type="password" name="userpwd" size="19"
            class="inputcss" style="background-color:#e8f4ff "
            onMouseOver="this.style.backgroundColor='#ffffff'"
            onMouseOut="this.style.backgroundColor='#e8f4ff'">
            </div></td>
    </tr>
    <tr>
        <td height="20"><div align="right">验证：</div></td>
        <td width="66" height="20">
            <div align="left"><input type="text" name="yz" size="10" class=
                "inputcss"
                style="background-color:#e8f4ff "
                onMouseOver="this.style.backgroundColor='#ffffff'"
                onMouseOut="this.style.backgroundColor='#e8f4ff'">
            </div></td>
        <td width="64"><div align="left">  
<?php
    $num=intval(mt_rand(1000,9999));
    for($i=0;$i<4;$i++){
        echo "<img src=images/code/".substr(strval($num),$i,1).".gif>";
    }
?>
</div></td>
    </tr>
    <tr>
        <td height="20" colspan="3"><div align="right">
            <input type="hidden" value="<?php echo $num;?>" name="num">
            <input name="submit" type="submit" class="buttoncss" value="提 交">
```

```
            <a href="agreereg.php">注册</a> <a href="javascript:openfindpwd()">
            找回密码</a> </div></td>
    </tr>
</form>
```

上述代码定义了一个用户登录表单，用户可以输入用户名、密码与验证码进行登录。单击【注册】按钮，将跳转到注册页面进行注册。单击【登录】按钮，将表单提交到用户登录验证页面 chkuser.php，该页面中验证用户身份的核心代码如下所示：

```
function checkinput(){
    include("conn/conn.php");
    $sql=mysql_query("select * from tb_user where name='".$this->name."'",$conn);
    $info=mysql_fetch_array($sql);
    if($info==false){
        echo "<script language='javascript'>alert('不存在此用户！');history.back();
            </script>";
        exit;
    }
    else{
        if($info[dongjie]==1){
           echo "<script language='javascript'>alert('该用户已经被冻结！ ');history.
                back(); </script>";
            exit;
        }
        if($info[pwd]==$this->pwd)
            session_start();
            $_SESSION[username]=$info[name];
            session_register("producelist");
            $producelist="";
            session_register("quatity");
            $quatity="";
            header("location:index.php");
            exit;
        }
        else {
            echo "<script language='javascript'>alert('密码输入错误!');history.
                back();</script>";
            exit;
        }
    }
}
```

上述代码定义的是一个验证用户身份是否合法的方法，chkuser.php 文件中获取用户登录表单提交过来的用户名与密码，通过这些数据并结合上述 checkinput()函数来验证用户的身份。登录成功之后会在页面中显示欢迎某用户的提示信息。

下面看看注册的实现，上述登录表单中已经提供了【注册】按钮，单击【注册】按钮将跳转到 agreereg.php 页面阅读注册协议，单击【我同意】链接将跳转到 reg.php 页面，在该页面中提供了一个用户注册表单，用户可以填写昵称、注册密码、电子邮件、QQ 号等信息进行注册，在此注册表单代码不再给出，仅给出用户输入信息的校验代码，如下所示：

```
<script language="javascript">
```

```
function chkinput(form){
    if(form.usernc.value==""){
        alert("请输入昵称!");
        form.usernc.select();
        return(false);
    }
    if(form.p1.value==""){
        alert("请输入注册密码!");
        form.p1.select();
        return(false);
    }
    if(form.p2.value==""){
        alert("请输入确认密码!");
        form.p2.select();
        return(false);
    }
    if(form.p1.value.length<6){
        alert("注册密码长度应大于 6!");
        form.p1.select();
        return(false);
    }
    if(form.p1.value!=form.p2.value){
        alert("密码与重复密码不同!");
        form.p1.select();
        return(false);
    }
    if(form.email.value==""){
        alert("请输入电子邮箱地址!");
        form.email.select();
        return(false);
    }
    if(form.email.value.indexOf('@')<0){
        alert("请输入正确的电子邮箱地址!");
        form.email.select();
        return(false);
    }
    if(form.tel.value==""){
        alert("请输入联系电话!");
        form.tel.select();
        return(false);
    }
    if(form.truename.value==""){
        alert("请输入真实姓名!");
        form.truename.select();
        return(false);
    }
    if(form.sfzh.value==""){
        alert("请输入身份证号!");
        form.sfzh.select();
        return(false);
    }
    if(form.dizhi.value==""){
        alert("请输入家庭住址!");
```

```
            form.dizhi.select();
            return(false);
        }
        if(form.tsda.value==""){
            alert("请输密码提示答案!");
            form.tsda.select();
            return(false);
        }
        if((form.ts1.value==1)&&(form.ts2.value=="")){
            alert("请选择或输入密码提示答案!");
            form.ts2.select();
            return(false);
        }
        return(true);
    }
</script>
```

上述代码依次取出用户提交的注册信息，进行判断校验是否合法。用户登录与注册页面执行效果如图 15-8 所示。注册成功之后将跳转到网站首页，用户可以在网站首页左边的登录表单中输入用户名与密码进行登录。



图 15-8 注册页面

15.5.4 用户中心

当用户单击首页【用户中心】链接时，可以进入用户中心，查看用户的个人信息、修改个人信息、进行用户留言、修改密码和注销离开网站。

1. 查看用户个人信息

在项目 shop/usercenter.php 文件中，定义了一个加载查看用户个人信息的表单，可以修改表单中的信息，单击【更改】按钮还可以更改用户的个人信息，代码如下所示：

```
<form name="form1" method="post" action="changeuser.php"
onsubmit="return chkinput1(this)">
<?php
    $sql=mysql_query("select * from tb_user where name='".$_SESSION[username]."'",
    $conn);                                   //查询用户的个人信息
    $info=mysql_fetch_array($sql);            //获取查询结果并将结果显示在表单中
        ?>
    <tr>
        <td width="100" height="20" bgcolor="#FFFFFF">
        <div align="center">昵称：</div></td>
        <td width="397" bgcolor="#FFFFFF"><div align="left">
        <?php echo $_SESSION[username];?></div></td>
    </tr>
    <tr>
        <td height="20" bgcolor="#FFFFFF">
        <div align="center">真实姓名：</div></td>
        <td height="20" bgcolor="#FFFFFF"><div align="left">
        <input type="text" name="truename" size="25" class="inputcssnull"
            value="
        <?php echo $info[truename];?>">
        </div></td>
    </tr>
    <tr>
        <td height="20" bgcolor="#FFFFFF"><div align="center">E-mail:
            </div></td>
        <td height="20" bgcolor="#FFFFFF"><div align="left">
        <input type="text" name="email" size="25" class="inputcssnull" value="
        <?php echo $info[email];?>">
        </div></td>
    </tr>
    <tr>
        <td height="20" bgcolor="#FFFFFF">
        <div align="center">QQ 号码：</div></td>
        <td height="20" bgcolor="#FFFFFF"><div align="left">
        <input type="text" name="qq" size="25" class="inputcssnull"
        value="<?php echo $info[qq];?>">
        </div></td>
    </tr>
    <tr>
    <td height="20" bgcolor="#FFFFFF"><div align="center">联系电话：</div></td>
        <td height="20" bgcolor="#FFFFFF"><div align="left">
        <input type="text" name="tel" size="25" class="inputcssnull" value="
        <?php echo $info[tel];?>">
        </div></td>
```

```
    </tr>
    <tr>
        <td height="20" bgcolor="#FFFFFF">
        <div align="center">家庭住址：</div></td>
        <td height="20" bgcolor="#FFFFFF"><div align="left">
        <input type="text" name="dizhi" size="25" class="inputcssnull" value="
            <?php echo $info[dizhi];?>"></div></td>
    </tr>
    <tr>
    <td height="20" bgcolor="#FFFFFF"><div align="center">邮政编码：</div></td>
        <td height="20" bgcolor="#FFFFFF"><div align="left">
        <input type="text" name="youbian" size="25" class="inputcssnull"
            value="<?php echo $info[youbian];?>"></div></td>
    </tr>
    <tr>
    <td height="20" bgcolor="#FFFFFF"><div align="center">身份证号：</div></td>
        <td height="20" bgcolor="#FFFFFF"><div align="left">
        <input type="text" name="sfzh" size="25" class="inputcssnull"
        value="<?php echo $info[sfzh];?>"></div></td>
    </tr>
    <tr>
        <td height="20" colspan="2" bgcolor="#FFFFFF"><div align="center">
        <input name="submit2" type="submit" class="buttoncss" value="更改">
          <input name="reset" type="reset" class="buttoncss"
        value="取消更改"></div></td>
    </tr>
</form>
```

上述代码实现了查看用户个人信息的同时还可以在表单中修改个人信息的功能，单击【更改】按钮，跳转到 changeuser.php 页面来获取更改的信息数据，并保存到数据库中。执行效果如图 15-9 所示。



图 15-9　查看用户个人信息

2. 修改个人信息

修改用户个人信息之后，单击【更改】按钮将表单更改信息提交到 changeuser.php 页面进行更改，更改完成页面将直接跳转到用户中心页面 usercenter.php，实现代码如下所示：

```
<?php
    include("conn/conn.php");   //引入连接数据库文件 conn.php，来连接数据库
    $email=$_POST[email];       //获取表单提交的电子邮件
    $truename=$_POST[truename]; //获取表单提交的用户真实姓名
    $sfzh=$_POST[sfzh];         //获取表单提交的身份证号
    $tel=$_POST[tel];           //获取表单提交的手机号
    $qq=$_POST[qq];             //获取表单提交的 QQ 号
    $dizhi=$_POST[dizhi];       //获取表单提交的地址
    $youbian=$_POST[youbian];   //获取表单提交的邮编
    mysql_query("update tb_user set email='$email',truename='$truename',sfzh=
'$sfzh',tel='$tel',
    qq='$qq',dizhi='$dizhi',youbian='$youbian'",$conn);
    header("location:usercenter.php");
?>
```

3. 用户留言

单击用户中心页面的【用户留言】链接，可以进入用户留言页面 userleaveword.php，该页面定义了一个用户留言表单，用户可以输入留言主题与内容，单击【提交】按钮，将留言数据提交到保存用户留言页面 saveuserleaveword.php，进行保存。保存留言的实现代码如下所示：

```
<?php
    session_start();                //开启 session
    $title=$_POST[title];           //获取表单提交的用户留言主题
    $content=$_POST[content];       //获取表单提交的用户留言内容
    $time=date("Y-m-j");            //获取当前时间为用户留言的时间
    include("conn/conn.php");       //引入连接数据库文件来连接数据库
    //查询留言用户信息数据
    $sql=mysql_query("select * from tb_user where name='".$_SESSION[username].
        "'",$conn);
    $info=mysql_fetch_array($sql);  //获取查询结果
    $userid=$info[id];              //取出留言用户的 id 号
    mysql_query("insert into tb_leaveword (title,content,time,userid)
        //将用户留言保存到数据库中
    values ('$title','$content','$time','$userid')",$conn);
    header("location:userleaveword.php");
        //用户留言保存完成后，页面跳转到用户留言页面
?>
```

上述代码先从表单中获取提交的用户留言主题与内容，并设置当前时间为用户留言时间，然后将用户留言保存到数据库中，最后页面跳转到用户留言页面。用户留言页面的执行效果如图 15-10 所示。

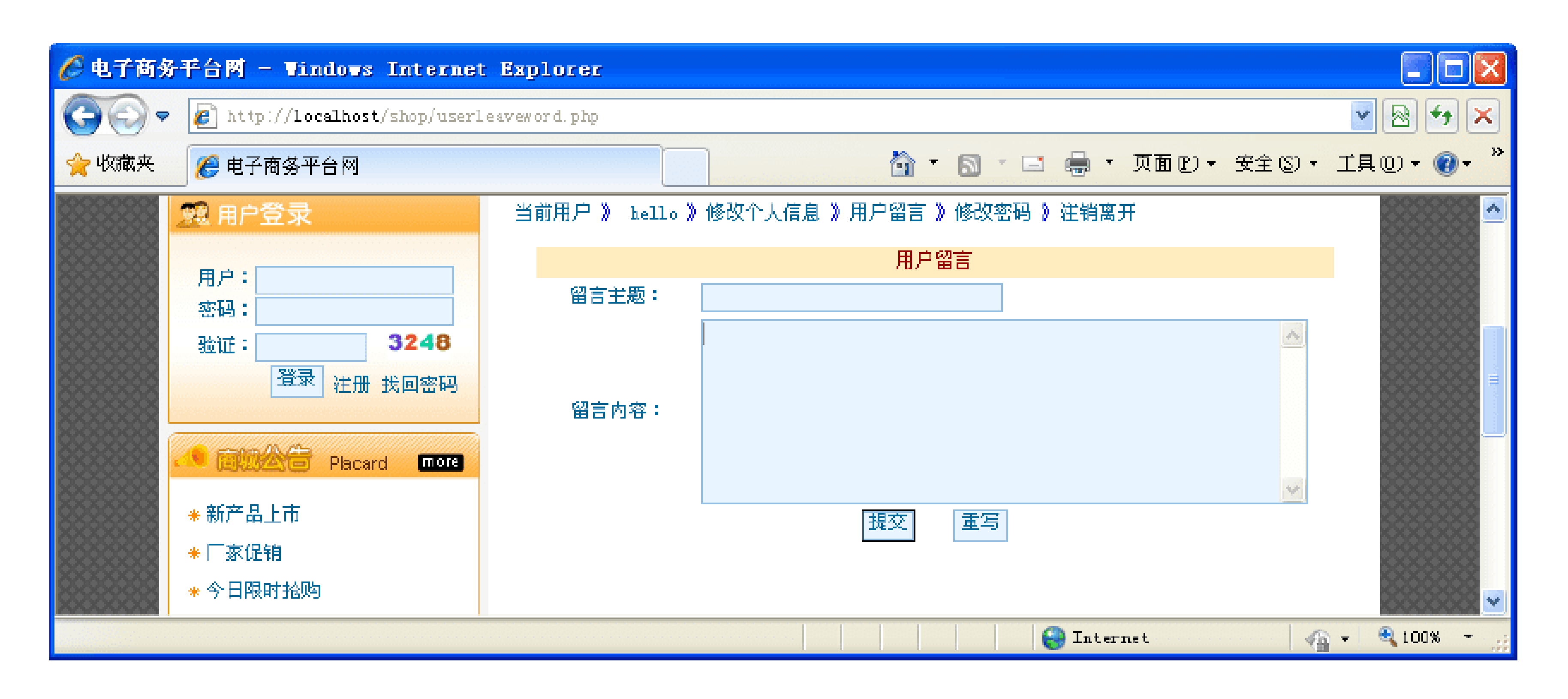


图 15-10　用户留言页面

4．修改密码

单击用户中心页面的【修改密码】链接，进入修改用户密码页面 changeuserpwd.php，该页面定义了一个修改用户密码表单，用户可以输入原密码、新密码和确认新密码，并单击【更改】按钮，将更改的密码数据提交到保存更改密码的页面 savechangeuserpwd.php，保存用户更改密码的实现代码如下所示：

```
<?php
    session_start();                        //开启 session
    $p1=md5(trim($_POST[p1]));              //获取表单提交的原密码
    $p2=trim($_POST[p2]);                   //获取表单提交的新密码
    $name=$_SESSION[username];              //从 session 中获取用户名
    class chkchange{                        //定义一个修改密码的类
        var $name;                          //用户名
        var $p1;                            //原密码
        var $p2;                            //新密码
        function chkchange($x,$y,$z){       //构造函数
            $this->name=$x;
            $this->p1=$y;
            $this->p2=$z;
        }
        function changepwd(){               //修改用户密码
            include("conn/conn.php");       //引入连接数据库文件 conn.php
            $sql=mysql_query("select * from tb_user where name='".$this->name.
                "'",$conn);
            $info=mysql_fetch_array($sql);  //查询获取用户原密码
            if($info[pwd]!=$this->p1){      //判断用户输入的原密码是否正确，输入错
                                              误提示用户错误信息
                echo "<script>alert('原密码输入错误!');history.back();</script>";
                exit;
            }
            else{           //输入原密码正确，执行修改密码 SQL 语句
                mysql_query("update tb_user set pwd='".md5($this->p2)."',
                    pwd1='$this->p2'
                where name='$this->name'",$conn);
```

```
                echo "<script>alert('密码修改成功!');history.back();</script>";
                exit;
            }
        }
    }
    $obj=new chkchange($name,$p1,$p2);
    $obj->changepwd()
?>
```

上述代码首先从用户更改密码表单中获取提交的原密码、新密码和从 session 中获取用户名，然后定义一个 chkchange 类，在该类中定义了用户名、原密码和新密码字段，一个带有 3 个参数的构造函数和一个更改密码的函数 changepwd()，最后，将从表单中获取的原密码、新密码和从 session 中获取的用户名传入构造函数创建一个 chkchange 类对象$obj，调用 changepwd()函数来更改密码。执行效果如图 15-11 所示。



图 15-11　修改用户密码

5．注销退出

单击用户中心页面的【注销退出】链接将直接退出该网站，页面将跳转到该网站的首页，该用户将只能浏览该网站商品，不能再执行查看个人信息、修改个人信息、购买商品等操作。实现代码如下所示：

```
<?php
    session_start();                //开启 session
    session_destroy();              //销毁 session 中的数据
    header("location:index.php");   //页面跳转到网站首页
?>
```

15.5.5 商品购买

用户登录进入网站之后，就可以执行购买商品操作了，可以单击商品右下角的【购买】图片链接将商品加入到购物车中，页面也将跳转到购物车页面，用户可以单击【继续购物】链接跳转到商品页面继续购买喜欢的商品，也可以单击【去收银台】生成订单结账付款，或者将某些不想要的商品从购物车中移除等。

实际上实现网上购物的主要环节就是将商品添加到购物车，下面就来看看如何将商品添加到购物车，代码如下所示：

```
<form name="form1" method="post" action="gouwu1.php">
    <tr>
        <td height="46" background="images/cart.gif"></td>
    </tr>
    <tr>
        <td bgcolor="#FFFFFF"><table width="500" border="0" align="center"
        cellpadding="0" cellspacing="1">
        <?php
            session_register("total");  //在 session 中注册一个 total 变量
            if($_GET[qk]=="yes"){        //判断 qk 是否等于 yes，等于 yes 将购物车清空
                $_SESSION[producelist]="";
                                        //在session 中定义一个producelist 变量，值为""
                $_SESSION[quatity]="";  //在 session 中定义一个 quatity 变量，值为""
            }
            $arraygwc=explode("@",$_SESSION[producelist]);//以@符号分割producelist
            $s=0;
            for($i=0;$i<count($arraygwc);$i++){
                $s+=intval($arraygwc[$i]);
                //依次从数据组取出数据通过 intval 转化为整型数据
            }
            if($s==0 ){       //为 0 提示购物车为空
                echo "<tr>";
                echo" <td height='25' colspan='6' bgcolor='#FFFFFF' align='center'>
                您的购物车为空!</td>";
                echo"</tr>";
            }else{
            ?>
    <tr>
        <td width="125" height="25" bgcolor="#FFFFFF"><div align="center">商
            品名称</div></td>
        <td width="52" bgcolor="#FFFFFF"><div align="center">数量</div></td>
        <td width="64" bgcolor="#FFFFFF"><div align="center">市场价</div></td>
        <td width="64" bgcolor="#FFFFFF"><div align="center">会员价</div></td>
        <td width="51" bgcolor="#FFFFFF"><div align="center">折扣</div></td>
        <td width="66" bgcolor="#FFFFFF"><div align="center">小计</div></td>
        <td width="71" bgcolor="#FFFFFF"><div align="center">操作</div></td>
    </tr>
```

```
    <?php
        $total=0;
        $array=explode("@",$_SESSION[producelist]);
        //将 producelist 以@符号分割为数组
        $arrayquatity=explode("@",$_SESSION[quatity]);
        //将 quatity 以@符号分割为数组
        while(list($name,$value)=each($_POST)){//each()和list()结合遍历数组$_POST
            for($i=0;$i<count($array)-1;$i++){
                if(($array[$i])==$name){
                    $arrayquatity[$i]=$value;
                }
            }
        }
        $_SESSION[quatity]=implode("@",$arrayquatity);
            //使用 implode()函数将数组$arrayquatity 以@符号作为间隔符连接为一个字符串
        for($i=0;$i<count($array)-1;$i++){
            $id=$array[$i];      //取出购买商品 id
            $num=$arrayquatity[$i]; //取出购买商品的数量
            if($id!=""){
                //查询购买商品的信息
                $sql=mysql_query("select * from tb_shangpin where id='".$id."'",$conn);
                $info=mysql_fetch_array($sql);  //获取购买商品的信息
                $total1=$num*$info[huiyuanjia];
                //取出购买商品的会员价计算购买商品的金额
                $total+=$total1;
                $_SESSION["total"]=$total;
                //将计算得到的购买商品的总金额放入 session 中
            ?>
    <tr>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info[mingcheng];?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
        <input type="text" name="<?php echo $info[id];?>" size="2" class=
            "inputcss"
            value=<?php echo $num;?>></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info[shichangjia];?>元</div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info[huiyuanjia];?>元</div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo @(ceil(($info[huiyuanjia]/$info[shichangjia])*100))."%";?>
                </div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info[huiyuanjia]*$num."元";?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <a href="removegwc.php?id=<?php echo $info[id]?>">移除</a></div>
</td>
    </tr>
    <?php
```

```
        }
    }?>
    <tr>
        <td height="25" colspan="8" bgcolor="#FFFFFF"><div align="right">
            <table width="500" height="25" border="0" align="center"
            cellpadding="0" cellspacing="0">
                <tr>
                    <td width="125"><div align="center">
                    <input name="submit2" type="submit" class="buttoncss"
                    value="更改商品数量"></div></td>
                    <td width="125"><div align="center"><a href="gouwu2.php">
                    去收银台</a></div></td>
                    <td width="125"><div align="center"><a href="index.php">
                    继续购物</a></div></td>    <!-- 单击继续购物页面跳转到首页继续浏览商
                                        品进行购物 -->
                    <td width="125"><div align="center"><a href="gouwu1.php?qk=yes">
                        清空购物车</a></div></td>
                    <td width="125"><div align="left">总计：<?php echo
                        $total;?></div></td>
                </tr>
            </table>
        </div></td>
    </tr>
    <?php
        }?>
    </table></td>
    </tr>
</form>
```

上述代码定义了一个表单，用来显示用户购买的商品信息，计算购买商品的金额，以及更改购买商品的数量、去收银台结账、继续购物、清空购物车等操作的链接，执行效果如图 15-12 所示。



图 15-12　购物车

15.5.6　去收银台结账

购买完商品之后，就要去收银台结账了，单击购物车页面中的【去收银台】链接，将跳转到收银台页面 gouwu2.php。该页面将要求用户填写收货人的信息，如收货人姓名、性别、邮政编码、联系电话、电子邮箱、详细地址等信息。填写完成之后可以单击【提交订单】按钮，发出订单。

1．去收银台

收银台页面 gouwu2.php 定义了一个填写收货人信息的表单，并使用 JavaScript 脚本验证用户输入的收货人信息是否合法，由于表单代码太多此处不全部给出，只给出表单输入数据验证代码，去收银台页面的详细实现代码可查看随书光盘本章目录中的 shop/gouwu2.php 文件。表单数据的验证代码如下所示：

```
<script language="javascript">
    function chkinput(form){
        if(form.name.value==""){
            alert("请输入收货人姓名!");
            form.name.select();
            return(false);
        }
        if(form.dz.value==""){
            alert("请输入收货人地址!");
            form.dz.select();
            return(false);
        }
        if(form.yb.value==""){
            alert("请输入收货人邮编!");
            form.yb.select();
            return(false);
        }
        if(form.tel.value==""){
            alert("请输入收货人联系电话!");
            form.tel.select();
            return(false);
        }
        if(form.email.value==""){
            alert("请输入收货人 E-mail 地址!");
            form.email.select();
            return(false);
        }
        if(form.email.value.indexOf("@")<0){
            alert("收货人 E-mail 地址格式输入错误!");
            form.email.select();
            return(false);
        }
        return(true);
```

```
    }
</script>
```

上述代码定义了一个验证用户输入收货人信息的函数 chkinput()，该函数依次从表单对象 form 中取出用户输入的收货人信息进行验证，合法则返回 true，否则返回 false。

如果验证用户输入的收货人信息完全合法，单击【提交订单】按钮，将信息提交给保存收货人信息页面 savedd.php，实现代码如下所示：

```
<?php
    session_start();                        //开启 session
    include("conn/conn.php");               //引入连接数据库文件 conn.php 来连接数据库
    $sql=mysql_query("select * from tb_user where name='".$_SESSION[username]."'",
$conn);
    $info=mysql_fetch_array($sql);              //查询下单人信息
    $dingdanhao=date("YmjHis").$info[id];       //通过当前时间生成一个订单号
    $spc=$_SESSION[producelist];                //获取购买商品列表
    $slc= $_SESSION[quatity];                   //获取购买商品数量
    $shouhuoren=$_POST[name2];                  //获取收货人姓名
    $sex=$_POST[sex];                           //获取收货人性别
    $dizhi=$_POST[dz];                          //获取收货人地址
    $youbian=$_POST[yb];                        //获取邮编
    $tel=$_POST[tel];                           //获取电话号码
    $email=$_POST[email];                       //获取电子邮件
    $shff=$_POST[shff];                         //获取送货方法
    $zfff=$_POST[zfff];                         //获取支付方法
    if(trim($_POST[ly])==""){                   //判断用户是否留言
        $leaveword="";                          //没有留言将$leaveword 变量赋值为""
    }else{
        $leaveword=$_POST[ly];                  //有留言将留言内容赋值给$leaveword 变量
    }
    $xiadanren=$_SESSION[username];             //获取下单人姓名
    $time=date("Y-m-j H:i:s");                  //设置下单时间
    $zt="未作任何处理";                         //初始化订单状态
    $total=$_SESSION[total];                    //设置该订单的总金额
    //发出 sql 语句将订单信息保存到数据库中
    mysql_query("insert into tb_dingdan(dingdanhao,spc,slc,shouhuoren,sex,
        dizhi,youbian,
    tel,email,shff,zfff,leaveword,time,xiadanren,zt,total)
     values
('$dingdanhao','$spc','$slc','$shouhuoren','$sex','$dizhi','$youbian','$tel
','$email','$shff',
    '$zfff','$leaveword','$time','$xiadanren','$zt','$total')",$conn);
    //保存订单完成之后，将跳转到显示订单信息页面 showdd.php
    header("location:showdd.php?dingdanhao=$dingdanhao");
?>
```

上述代码首先从表单中获取用户输入的订单信息，然后发出 SQL 语句将订单信息保存到数据库中，这样订单就生成了，执行效果如图 15-13 所示。



图 15-13　生成订单

2．显示订单信息

单击【提交订单】按钮保存用户订单信息，成功之后页面将自动跳转到显示订单信息页面 showdd.php，以供用户查看自己订单的信息，该页面首先查询了订单信息，然后查询了用户购买商品的信息，最后在一个表格中显示用户订单的信息与购买商品的信息。在此只给出查询订单与用户购买商品信息的代码，不提供显示订单与商品信息的表格实现代码。查询订单信息的代码如下所示：

```
<?php
    include("conn/conn.php");          //引入连接数据库文件 conn.php
    $dingdanhao=$_GET[dingdanhao];  //获取订单号
    //根据订单号查询该订单信息
   $sql=mysql_query("select * from tb_dingdan where dingdanhao='".$dingdanhao.
        "'",$conn);
    $info=mysql_fetch_array($sql);  //获取查询订单结果
    $spc=$info[spc];                 //获取购买商品的编号
    $slc=$info[slc];                 //获取购买商品的数量
    $arraysp=explode("@",$spc);      //以@符号为标记，分割字符串$spc
    $arraysl=explode("@",$slc);      //以@符号为标记，分割字符串$slc
?>
```

上述代码通过订单号查询订单信息，并依次取出查询结果中的订单信息，下面来看看如何查询用户购买的商品信息，代码如下所示：

```
<?php
    $total=0;          //定义商品总金额变量
    for($i=0;$i<count($arraysp)-1;$i++){
```

```
        if($arraysp[$i]!=""){          //从$arraysp 数组中依次取出购买商品的 id 号
        //通过商品 id 号查询商品的信息
        $sql1=mysql_query("select * from tb_shangpin where id='".$arraysp[$i].
            "'",$conn);
        $info1=mysql_fetch_array($sql1);                //获取商品查询结果
        $total=$total+=$arraysl[$i]*$info1[huiyuanjia];//计算购买商品的总金额
?>
```

上述代码定义了一个存放商品总金额的变量，然后依次从$arraysp 数组中取出购买商品的 id 号，根据商品 id 号查询商品的信息，并计算出购买商品的总金额。显示订单信息页面的执行效果如图 15-14 所示。



图 15-14　显示订单信息

15.6　后台数据管理

本网站后台是针对网站管理员设计的，主要包含商品管理、用户管理、订单管理和评论管理 4 个大模块，本节将针对这 4 个主要模块详细讲解网站后台管理的实现。

15.6.1　商品管理

管理员登录网站后台之后，可以对网站商品进行管理，如添加商品、修改商品、商品类别管理、添加商品类别。下面先来看看如何实现商品的添加。

1．添加商品

首先在商品添加页面 addgoods.php 中定义了一个表单，管理员可以在该表单中填写添加商品的信息，如商品名称、上市时间、市场价、会员价、商品类型、商品等级、品牌、上传商品图片等。单击表单的【添加】按钮，将添加商品数据提交给保存商品页面 savenewgoods.php 进行保存，实现代码如下所示：

```
<?php
```

```
include("conn/conn.php");   //引入连接数据库文件 conn.php
//从表单中取出市场价与会员价判断管理员输入数据是否合法
if(is_numeric($_POST[shichangjia])==false || is_numeric($_POST
    [huiyuanjia])==false){
    echo "<script>alert('价格只能为数字！');history.back();</script>";
    exit;
}
if(is_numeric($_POST[shuliang])==false){        //判断管理员输入数量是否合法
    echo "<script>alert('数量只能为数字！');history.back();</script>";
    exit;
}
$mingcheng=$_POST[mingcheng];                   //获取商品名称
$nian=$_POST[nian];                             //获取商品上市年份
$yue=$_POST[yue];                               //获取商品上市月份
$ri=$_POST[ri];                                 //获取商品上市日
$shichangjia=$_POST[shichangjia];               //获取商品市场价
$huiyuanjia=$_POST[huiyuanjia];                 //获取会员价
$typeid=$_POST[typeid];                         //获取商品类型
$dengji=$_POST[dengji];                         //获取商品等级
$xinghao=$_POST[xinghao];                       //获取商品型号
$pinpai=$_POST[pinpai];                         //获取商品品牌
$tuijian=$_POST[tuijian];                       //获取商品是否为推荐商品
$shuliang=$_POST[shuliang];                     //获取商品数量
$upfile=$_POST[upfile];                         //获取商品图片
if(ceil(($huiyuanjia/$shichangjia)*100)<=80){   //判断商品是否为特价商品
    $tejia=1;
}else{
    $tejia=0;
}
function getname($exname){                      //获取商品上传图片的名字
    $dir = "upimages/";
    $i=1;
    if(!is_dir($dir)){
        mkdir($dir,0777);
    }
    while(true){
        if(!is_file($dir.$i.".".$exname)){
            $name=$i.".".$exname;
            break;
        }
        $i++;
    }
    return $dir.$name;
}
//获取上传图片的路径
$exname=strtolower(substr($_FILES['upfile']['name'],(strrpos($_FILES['u
pfile']['name'],'.')+1)));
$uploadfile = getname($exname);     //获取商品上传图片的名字
move_uploaded_file($_FILES['upfile']['tmp_name'], $uploadfile);
```

```
    //上传的文件移动到新位置。若成功，则返回 true，否则返回 false
    if(trim($_FILES['upfile']['name']!="")){
        $uploadfile="admin/".$uploadfile;
    }else{
        $uploadfile="";
    }
    $jianjie=$_POST[jianjie];                    //获取商品简介
    $addtime=$nian."-".$yue."-".$ri;             //设置商品的添加时间
    //发出 SQL 语句将添加商品信息保存到数据库中
    mysql_query("insert into tb_shangpin(mingcheng,jianjie,addtime,dengji,
        xinghao,tupian,typeid,shichangjia,huiyuanjia,pinpai,tuijian,
        shuliang,cishu)values('$mingcheng','$jianjie','$addtime','$dengji',
        '$xinghao','$uploadfile','$typeid','$shichangjia','$huiyuanjia',
        '$pinpai','$tuijian','$shuliang','0')",$conn);
    //输出提示商品添加成功，页面跳转到添加商品页面 addgoods.php
    echo "<script>alert('商品".$mingcheng."添加成功!');window.location.href=
'addgoods.php';</script>";
?>
```

上述代码从添加商品页面 addgoods.php 获取用户输入的商品信息，并将上传的商品图片移动到新的位置，发出 insert 语句将添加的商品信息保存到数据库中。执行效果如图 15-15 所示。



图 15-15　添加商品

2．修改商品

单击后台主页左边的【修改商品】链接即可进入修改商品页面 editgoods.php，首先看到的是本网站所有商品的列表信息，用户可以启用商品记录旁边的复选框，对商品信息数据进行更改与删除操作，当用户选择一条商品记录后单击【更改】链接时，将跳转到更改商品信息页面 changegoods.php，该页面定义了一个商品信息表单，并将选中要修改的商品信息数据加载显示在表单中，用户可以修改表单中的数据，单击表单【更改】按钮，将更改后的数据提交到保存

商品更改信息页面 savechangegoods.php，该页面的实现与保存添加商品页面 savenewgoods.php 非常相似，在此代码不再给出。修改商品页面的实现代码如下所示：

```
<?php
    include("conn/conn.php");        //引入连接数据库文件 conn.php
?>
<body topmargin="0" leftmargin="0" bottommargin="0">
<?php
    $sql=mysql_query("select count(*) as total from tb_shangpin ",$conn);
    //查询获取商品总数
    $info=mysql_fetch_array($sql); //获取查询结果
    $total=$info[total];     //将查询的商品总数赋值给$total 变量
    if($total==0){       //判断商品总数如果为 0，显示“本站暂无商品!”
        echo "本站暂无商品!";
    }else{            //否则的话，在表单中显示本站所有商品的信息
?>
<form name="form1" method="post" action="deletefxhw.php"><p> </p>
    <table width="750" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
        <td height="75" bgcolor="#666666"><table width="750" height="86"
        border="0" cellpadding="0" cellspacing="1">
        <tr bgcolor="#FFCF60">
            <td height="20" colspan="10"><div align="center" class="style1">
                商品信息编辑</div></td>
    </tr>
    <tr>
        <td width="59" height="28" bgcolor="#FFFFFF"><div align="center">复选
            </div></td>
        <td width="102" bgcolor="#FFFFFF"><div align="center">名称</div></td>
        <td width="86" bgcolor="#FFFFFF"><div align="center">品牌</div></td>
        <td width="71" bgcolor="#FFFFFF"><div align="center">型号</div></td>
        <td width="60" bgcolor="#FFFFFF"><div align="center">剩余</div></td>
        <td width="60" bgcolor="#FFFFFF"><div align="center">市场价</div></td>
        <td width="61" bgcolor="#FFFFFF"><div align="center">会员价</div></td>
        <td width="60" bgcolor="#FFFFFF"><div align="center">卖出</div></td>
        <td width="112" bgcolor="#FFFFFF"><div align="center">加入时间</div>
</td>
        <td width="68" bgcolor="#FFFFFF"><div align="center">操作</div></td>
    </tr>
    <?php
        $pagesize=20;        //设置本页显示 20 个商品
        if ($total<=$pagesize){//如果本站商品小于等于 20，设置显示页数为 1
            $pagecount=1;
        }
        if(($total%$pagesize)!=0){ //判断商品总数是否能被 20 整除
            $pagecount=intval($total/$pagesize)+1;     //计算商品显示页数
        }else{
            $pagecount=$total/$pagesize;    //计算商品显示页数
        }
```

```
        if(($_GET[page])==""){                   //判断 page 参数是否为""
            $page=1;
        }else{
            $page=intval($_GET[page]);           //获取 page 参数转换为整型数据
        }
        $sql1=mysql_query("select * from tb_shangpin order by addtime desc
        limit ".($page-1)*$pagesize.",$pagesize",$conn);
        //以分页形式查询商品的所有信息
        while($info1=mysql_fetch_array($sql1)){
        //通过 while 循环在表格中分页显示所有商品的信息
    ?>
    <tr>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <input type="checkbox" name="<?php echo $info1[id];?>" value=
                <?php echo $info1[id];?>></div></td>
        <td height="25" bgcolor="#FFFFFF">
            <div align="center"><?php echo $info1[mingcheng];?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info1[pinpai];?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info1[xinghao];?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center"><?php if($info1
            [shuliang]<0)
            {echo "售完";}else {echo $info1[shuliang];}?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info1[shichangjia];?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info1[huiyuanjia];?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info1[cishu];?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <?php echo $info1[addtime];?></div></td>
        <td height="25" bgcolor="#FFFFFF"><div align="center">
            <a href="changegoods.php?id=<?php echo $info1[id];?>">更改</a>
</div></td>
        </tr><?php}?>
    </table></td>
    </tr>
</table>
```

上述代码可以查询出本站所有商品的信息，并通过分页将所有商品信息显示在表格中，还为每条商品记录提供了【更改】商品链接。下面再提供一个【删除选择】商品链接和分页显示商品工具条，用户可以通过单击分页工具条依次翻看商品信息的数据，代码如下所示：

```
<table width="750" height="25" border="0" align="center" cellpadding="0"
cellspacing="0">
    <tr>
        <td width="165"><div align="left"><input name="submit" type="submit"
            class="buttoncss" id="submit" value="删除选择">
```

```
             <input type="reset" value="重新选择" class="buttoncss"></div>
</td>
        <td width="585"><div align="right"> 本站共有货物
            <?php echo $total;?>
             件 每页显示 <?php echo $pagesize;?>
             件 第 <?php echo $page;?> 页/共 
            <?php echo $pagecount; ?> 页
            <?php
                if($page>=2){?>
            <a href="editgoods.php?page=1" title="首页"><font face="webdings">
9 </font></a>
            <a  href="editgoods.php?id=<?php  echo  $id;?>&page=<?php  echo
$page-1;?>"
            title="前一页"><font face="webdings"> 7 </font></a>
            <?php
                }if($pagecount<=4){
                    for($i=1;$i<=$pagecount;$i++){
            ?>
            <a href="editgoods.php?page=<?php echo $i;?>"><?php echo $i;?></a>
            <?php
                }
            }else{
                for($i=1;$i<=4;$i++){
            ?>
            <a href="editgoods.php?page=<?php echo $i;?>"><?php echo $i;?></a>
            <?php }?>
            <a href="editgoods.php?page=<?php echo $page-1;?>" title="后一页
">
            <font face="webdings"> 8 </font></a> <a href="editgoods.php?id=
<?php echo $id;?>
            &page=<?php echo $pagecount;?>" title="尾页"><font face="webdings"> :
</font></a>
            <?php }?>
            </div></td>
        </tr>
        </table>
        <p> </p>
    </form>
    <?php
    }
?>
```

上述代码实现了删除商品的功能，用户选择一条商品记录后单击【删除选择】按钮即可删除商品，最后还实现了商品的分页显示，执行效果如图 15-16 所示。

商品信息编辑

复选	名称	品牌	型号	剩余	市场价	会员价	卖出	加入时间	操作
☑	女童连衣裙	lianyiqun	lyq	245	239	149	0	2010-6-7	更改
☐	时尚女士连衣裙	lianyiqun	lyq	475	589	456	0	2010-6-2	更改
☐	长款羽绒服	yurongfu	yrf	388	685	458	0	2010-12-15	更改
☐	家庭影院	时尚LOVE	LOVE-99	98	9999	9599	1	2007-1-1	更改
☐	液晶显示器LVOE	LOVE2	LOVE-2	88	1499	1399	0	2007-1-1	更改
☐	经美壁纸	a	a	200	1000	800	10	2007-1-1	更改
☐	数码相机	时尚	XM-5001	20000	2699	2599	10	2006-5-5	更改
☐	液晶显示器	时尚LOVE	LOVE-999	99	2999	2666	0	2006-1-1	更改
☐	全自动洗衣机	X44	301X	8000	1800	1699	0	2005-7-8	更改
☐	玩具车	LOVE-66	Handle-66	666	399	366	0	2002-2-1	更改
☐	水杯	44	44	1800	100	98	22	1995-1-1	更改

删除选择　重新选择　本站共有货物 11 件 每页显示 20 件 第 1 页/共 1 页 1

图 15-16　修改商品

商品管理模块还实现了商品类别管理和添加商品类别，这两个功能的实现与上述讲到的添加商品和修改商品功能实现的核心方法一样，因此就不再详细讲解，大家可以查看随书光盘本章目录中的 shop/admin 目录下的 showleibie.php 文件来学习商品类别管理的实现，查看 addleibie.php 和 saveaddleibie.php 文件学习添加商品类别和保存添加商品类别的实现。

15.6.2　用户管理

用户管理与商品管理还有些不同，管理员不可以随意添加用户，但可以删除用户并查看用户的信息，以及查看删除用户的留言，还可以更改管理员的信息，比如管理员名称、密码等。下面就来看看这些功能的详细实现吧。

1．用户信息管理

单击后台主页左边的【用户信息管理】链接，就可以看到用户信息管理页面 edituser.php。该页面首先查询获取本网站用户的总数，计算出分页显示的页数，再分页查询用户的详细信息，显示在表格中，最后提供了查询、删除和分页显示用户的分页工具条。实现方式与修改商品页面 editgoods.php 一样，因此不再给出代码，详细实现代码可查看项目源码 shop/admin/文件夹下的 edituser.php 文件。执行效果如图 15-17 所示。

当用户选中表格中的用户记录后单击【删除选项】即可删除该用户记录，其实现代码如下所示：

```
<?php
    include("conn/conn.php");   //引入连接数据库文件 conn.php
    while(list($name,$value)=each($_POST)){//通过 while 循环依次取出
        mysql_query("delete from tb_user where id=".$value."",$conn);
        mysql_query("delete from tb_pingjia where userid=".$value."");
        mysql_query("delete from tb_leaveword where userid=".$value."",$conn);
    }
```

```
    header("location:edituser.php");          //删除成功后页面跳转到 edituser.php
?>
```

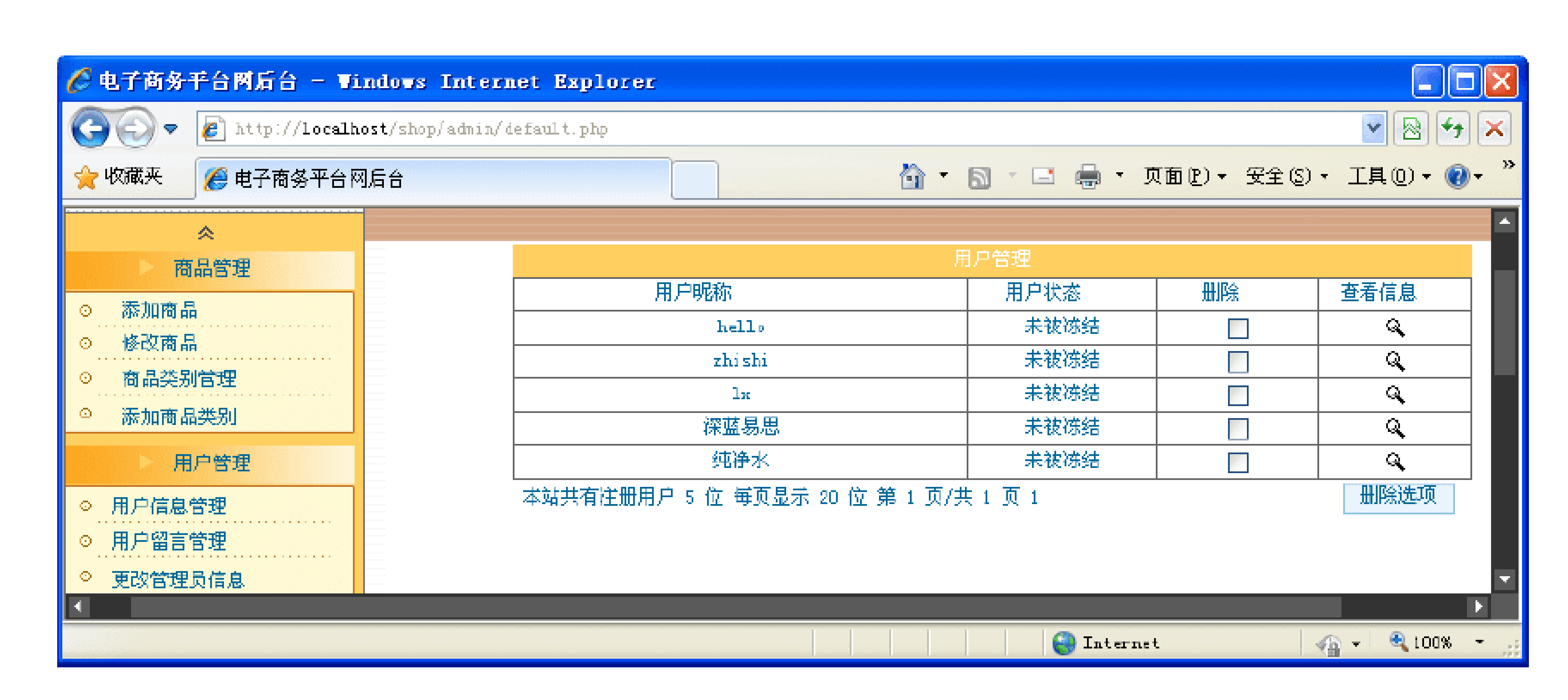


图 15-17　用户信息管理

上述代码通过 while 循环删除了被选中的所有的用户信息，以及与用户相关的评价和留言信息。删除成功后页面跳转到用户信息管理页面 edituser.php。

2．用户留言管理

单击后台主页左边的【用户留言管理】链接，可以在页面中央看到用户的留言信息列表，选中列表中的留言后单击【查看】操作链接，即可查看用户留言的详细内容。执行效果如图 15-18 所示。

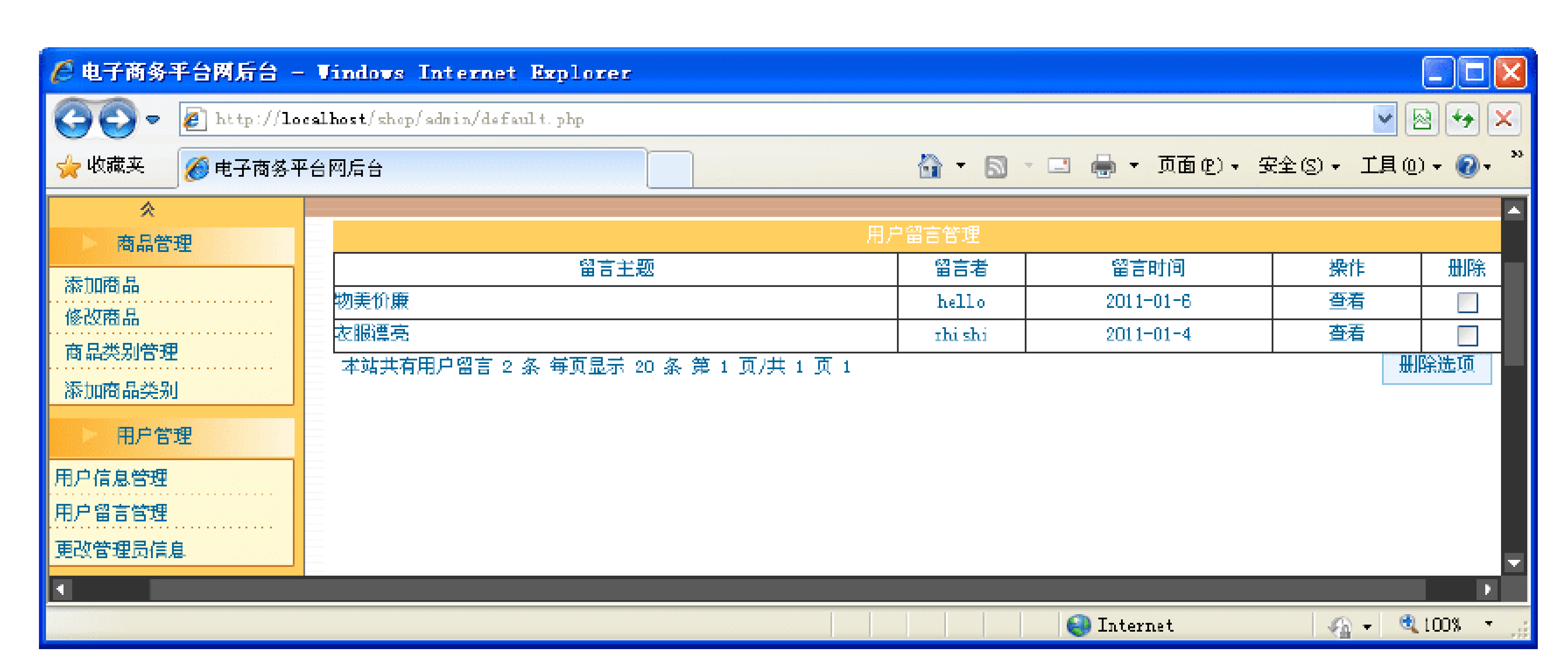


图 15-18　用户留言管理

用户留言管理页面的实现方式与用户信息管理大致相同，在此代码省略，详细代码实现可查看随书光盘本章目标中的 shop/admin/lookleaveword.php 文件。查看用户留言代码的实现如下所示。

```
<?php
    include("conn/conn.php");          //引入连接数据库文件 conn.php
    $id=$_GET[id];                     //获取用户留言 id
    $sql=mysql_query("select * from tb_leaveword where id='".$id."'",$conn);
                                       //通过留言 id 查询留言详细内容
    $info=mysql_fetch_array($sql);     //获取查询结果
```

```
?>
```

上述代码通过用户留言 id 查询用户留言的详细内容，并将查询到的留言主题、时间和内容显示在查看用户留言页面 editleaveword.php 定义的表格中。执行效果如图 15-19 所示。

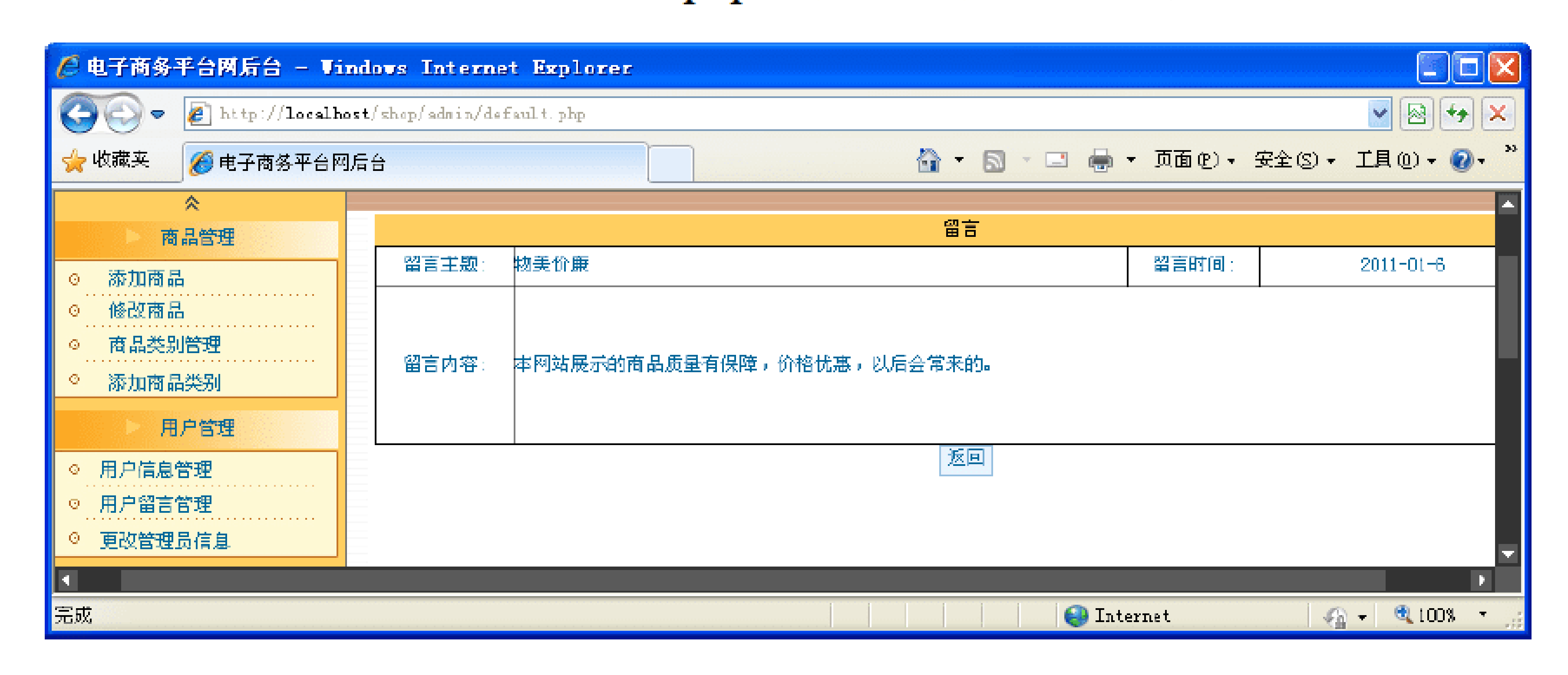


图 15-19　查看留言

不仅可以查看留言，选择用户留言，单击用户留言管理页面中的【删除选项】按钮还可删除用户留言。具体实现代码如下所示：

```
<?php
    include("conn/conn.php");                    //引入连接数据库文件 conn.php
    while(list($name,$value)=each($_POST))       //通过循环依次迭代删除选中的用户留言信息
    {
        mysql_query("delete from tb_leaveword where id='".$value."'",$conn);
    }
    header("location:lookleaveword.php");        //删除留言之后页面跳转到用户留言
                                                 //管理页面 lookleaveword.php
?>
```

上述代码引入 conn.php 文件连接数据库，通过 while 循环依次迭代删除选中的用户留言信息，删除成功后将页面重新跳转到用户留言管理页面 lookleaveword.php。

更改管理员模块与修改密码的实现方式大致相同，在此代码省略，详细代码的实现可查看随书光盘本章目录中的 shop/admin/目录下的 changeadmin.php 和 savechangeadmin.php 文件。

15.6.3　订单管理

订单管理也是一个电子商务网站后台管理十分重要的模块，管理员登录后台管理之后，单击后台主页左边订单管理下面的【编辑订单】链接，即可看到本网站用户提交的所有订单的列表，单击【查看】按钮可查看订单的详细信息，单击【执行】按钮可以修改订单的状态，如已发货、已收货、已收款等。在列表中启用订单记录的复选框，单击列表底部的【删除选项】按钮即可删除订单记录。订单管理模块的详细讲解如下。

1. 查看订单

在编辑订单页面中，单击订单列表中的【查看】链接，即可查看订单的详细信息，如订单

号、购买商品列表、商品的总计费用、下单人、收货人等。具体实现代码如下所示：

```
<?php
    include("conn/conn.php");   //引入连接数据库文件 conn.php
    $id=$_GET[id];              //获取要查询订单的 id
    $sql=mysql_query("select * from tb_dingdan where id='".$id."'",$conn);
    //通过订单 id 查询订单详细信息
    $info=mysql_fetch_array($sql); //获取查询结果
    $spc=$info[spc];               //取出购买商品的编号
    $slc=$info[slc];               //取出购买商品的数量
    $arraysp=explode("@",$spc);    //将购买商品编号字符串按@符号分割为一个数组
    $arraysl=explode("@",$slc);    //将购买商品数量字符串按@符号分割为一个数组
?>
```

下面代码将上述代码查询到的订单信息数据显示在表格中：

```
<body topmargin="0" leftmargin="0" bottommargin="0">
    <table width="600"  border="0" align="center" cellpadding="0"
        cellspacing="0">
        <tr align="center" bgcolor="#FFCF60">
            <td height="20" colspan="2">商品订单</td>
        </tr>
        <tr>
            <td width="448" height="20">订单号：<?php echo $info[dingdanhao];?>
</td>
            <td width="152"><div align="right">
    <script>
        function prn(){
            document.all.WebBrowser1.ExecWB(7,1);
        }
    </script>
    <object ID='WebBrowser1' WIDTH=0
    HEIGHT=0 CLASSID='CLSID:8856F961-340A-11D0-A96B-00C04FD705A2'></object>
        <input type="button" value="打印预览" class="buttoncss" onClick="prn()">
             
        <input type="button" value="打印" class="buttoncss" onClick="window.
print()"></div></td>
    </tr>
    <tr>
        <td height="20" colspan="2">商品列表(如下)：</td>
    </tr>
</table>
<table width="500" height="60" border="0" align="center" cellpadding="0"
cellspacing="0">
    <tr>
        <td bgcolor="#666666"><table width="500" border="0" align="center"
        cellpadding="0" cellspacing="1">
    <tr bgcolor="#FFCF60">
        <td width="153" height="20">商品名称</td>
        <td width="80">市场价</td>
```

```
        <td width="80">会员价</td>
        <td width="80">数量</td>
        <td width="101">小计</td>
    </tr>
    <?php
        $total=0;
        for($i=0;$i<count($arraysp)-1;$i++){
        if($arraysp[$i]!=""){
            $sql1=mysql_query("select * from tb_shangpin where id='".$arraysp[$i].
                "'",$conn);
            $info1=mysql_fetch_array($sql1);
            $total=$total+=$arraysl[$i]*$info1[huiyuanjia];
    ?>
    <tr bgcolor="#FFFFFF">
        <td height="20"><?php echo $info1[mingcheng];?></td>
        <td height="20"><?php echo $info1[shichangjia];?></td>
        <td height="20"><?php echo $info1[huiyuanjia];?></td>
        <td height="20"><?php echo $arraysl[$i];?></td>
        <td height="20"><?php echo $arraysl[$i]*$info1[huiyuanjia];?></td>
    </tr>
    <?php
            }
        }
    ?>
    <tr bgcolor="#FFFFFF">
        <td height="20" colspan="5">总计费用:<?php echo $total;?></td>
    </tr>
    </table></td>
    </tr>
    </table>
    <table width="460" border="0" align="center" cellpadding="0" cellspacing="0">
        <tr>
            <td width="81" height="20">下单人：</td>
            <td colspan="3"><?php echo $info[xiadanren];?></td>
        </tr>
        <tr>
            <td height="20">收货人：</td>
            <td height="20" colspan="3"><?php echo $info[shouhuoren];?></td>
        </tr>
        <tr>
            <td height="20">收货人地址：</td>
            <td height="20" colspan="3"><?php echo $info[dizhi];?></td>
        </tr>
        <tr>
            <td height="20">邮  编：</td>
            <td width="145" height="20"><?php echo $info[youbian];?></td>
            <td width="66">电  话：</td>
            <td width="158"><?php echo $info[tel];?></td>
        </tr>
```

```
        <tr>
            <td height="20">E-mail:</td>
            <td height="20"><?php echo $info[email];?></td>
            <td height="20"> </td>
            <td height="20"> </td>
        </tr>
        <tr>
            <td height="20">送货方式：</td>
            <td height="20"><?php echo $info[shff];?></td>
            <td height="20">支付方式：</td>
            <td height="20"><?php echo $info[zfff];?></td>
        </tr>
        <tr>
            <td height="20" colspan="4"><span class="style3">请您在一周内按您的
                支付方式进行汇款,汇款时注明您的订单号!汇款后请及时通知我们</span></td>
        </tr>
    <tr>
        <td height="20"> </td>
        <td height="20"><div align="center"><input type="button" onClick=
            "window.close()"
            value="关闭窗口" class="buttoncss"></div></td>
        <td height="20">创建时间：</td>
        <td height="20"><?php echo $info[time];?></td>
    </tr>
</table>
```

查询订单信息的执行效果如图 15-20 所示。



图 15-20　查询订单

2. 执行处理订单

在编辑订单页面中，当用户单击列表中的【执行】按钮时即可进入订单执行处理页面，从中可以设置订单状态为已收款、已发货或已收货，单击【修改】按钮将完成订单处理操作。执行效果如图 15-21 所示。



图 15-21　订单处理

订单处理页面 orderdd.php 通过订单 id 号，查询到该订单的详细信息，定义了一个表格来显示订单的详细信息，还在表格中定义了一个执行订单处理的表单，当管理员在该表单中启用复选框修改订单状态后，单击【修改】按钮即可将修改数据提交到保存订单处理页面 saveorder.php，具体实现代码如下所示：

```
<?php
    $ysk=$_POST[ysk]." ";          //获取是否已收款
    $yfh=$_POST[yfh]." ";          //获取是否已发货
    $ysh=$_POST[ysh]." ";          //获取是否已送货
    $zt="";                             //初始订单状态为""
    if($ysk!=" "){                 //判断是否设置了订单状态为已收款
        $zt.=$ysk;
    }
    if($yfh!=" "){                 //判断是否设置了订单状态为已发货
        $zt.=$yfh;
    }
    if($ysh!=" "){                 //判断是否设置订单状态为已送货
        $zt.=$ysh;
    }
    if(($ysk==" ")&&($yfh==" ")&&($ysh==" ")){
    //判断若三种订单状态都未设置将提示选择处理状态
        echo "<script>alert('请选择处理状态!');history.back();</script>";
        exit;
    }
    include("conn/conn.php");                   //引入连接数据库文件 conn.php
    $sql3=mysql_query("select * from tb_dingdan where id='".$_GET[id]."'",$conn);
                                                //查询订单信息
    $info3=mysql_fetch_array($sql3);            //获取查询结果
    if(trim($info3[zt])=="未作任何处理"){       //判断订单状态是否为未做任何处理
```

```
        $sql=mysql_query("select * from tb_dingdan where id='".$_GET[id]."'",
            $conn);
        $info=mysql_fetch_array($sql);
        $array=explode("@",$info[spc]);//将查询到的商品编号字符串分割为一个数组
        $arraysl=explode("@",$info[slc]);//将查询到的商品数量字符串分割为一个数组
        for($i=0;$i<count($array);$i++){    //通过 for 依次取出商品编号和购买数量，
            修改商品表中商品浏览次数和商品数量
            $id=$array[$i];
            $num=$arraysl[$i];
            mysql_query("update tb_shangpin set cishu=cishu+'".$num."' ,
            shuliang=shuliang-'".$num."' where id='".$id."'",$conn);
        }
    }
    //修改订单的状态
    mysql_query("update tb_dingdan set zt='".$zt."'where id='".$_GET[id].
        "'",$conn);
    header("location:lookdd.php");
?>
```

上述代码首先获取表单提交的状态，判断确认管理员设置的订单状态，然后通过订单号查询获取订单购买商品的编号和数量信息，使用 for 循环依次取出购买商品的编号和数量，进而修改商品表中商品的浏览次数和商品数量，最后修改订单的状态，修改成功后将页面重新跳转到查看订单页面 lookdd.php。

3. 删除订单

当一些订单交易完全结束时，管理员即可进入后台将这些订单信息删除。在查看订单页面的订单列表中，启用需要删除的订单记录的复选框后单击【删除选项】即可完成删除操作。具体实现代码如下所示：

```
<?php
    $page=intval($_POST[page_id]);  //获取选中订单记录的页数，并转换为整型
    include("conn/conn.php");       //引入连接数据库文件 conn.php
    while(list($value,$name)=each($_POST)) //通过 while 依次迭代删除选中的订单记录
    {
        mysql_query("delete from tb_dingdan where id='".$value."'",$conn);
    }
    header("location:lookdd.php?page=".$page."");
        //删除完成之后，重新跳转到查看订单信息页面
?>
```

上述代码引入连接数据库文件 conn.php，通过 while 循环依次迭代删除选中的订单记录，删除成功后，将页面重新跳转到查看订单信息页面。

15.6.4　信息管理

信息管理模块主要是对本网站的一些公告信息和用户对该网站商品的评论信息的管理，公告信息是指本网站的一些商品促销活动、商品特卖、新产品的推出等。管理员通过后台将这些

信息添加到数据库，用户登录到网站之后就可以看到本网站近期商品的一些活动，帮助用户买到更加物美价廉的商品。

管理员登录后台主页之后，可以单击页面左边的【公告管理】链接，进入公告管理页面，在该页面中定义了一个表格用来加载显示当前网站所有的公告信息数据，还提供了【查看编辑】链接和【删除所选项】按钮，使管理员可以查看和删除公告信息。执行效果如图 15-22 所示。



图 15-22 公告管理

由于该页面的实现与用户留言管理页面、编辑订单页面大致相同，因此不再进行详细讲解，详细代码实现可查看随书光盘本章目录中的源码 shop/admin/admingonggao.php 文件。下面看看如何实现公告信息的查看编辑和删除功能。

1. 查看编辑公告

单击公告管理页面中的【查看编辑】链接，将进入查看编辑公告页面，该页面定义了一个表格，在表格中定义了一个表单，将查询到的公告信息加载显示到表单中，也可以修改表单中的公告信息数据，单击【更改】按钮，将修改的公告信息提交给保存公告信息页面 saveeditgonggao.php 进行保存。具体实现代码如下所示：

```
<?php
    $title=$_POST[title];              //获取公告标题
    $content=$_POST[content];          //获取公告内容
    include("conn/conn.php");          //引入连接数据库文件 conn.php
    mysql_query("update tb_gonggao set title='$title',content='$content'
    where id='".$_POST[id]."'",$conn);      //发出 update 语句修改公告
    echo "<script>alert('公告修改成功!');history.back();</script>";
                                            //提示公告修改成功
?>
```

上述代码从查看编辑公告页面 editgonggao.php 获取了提交的修改公告信息数据，发出 update 语句修改公告信息。查看编辑公告页面的实现方法与查看订单实现方式大致相同，因此代码不再在给出，详细代码实现可查看随书光盘本章目录中的 shop/admin/editgonggao.php 文

件。执行效果如图 15-23 所示。

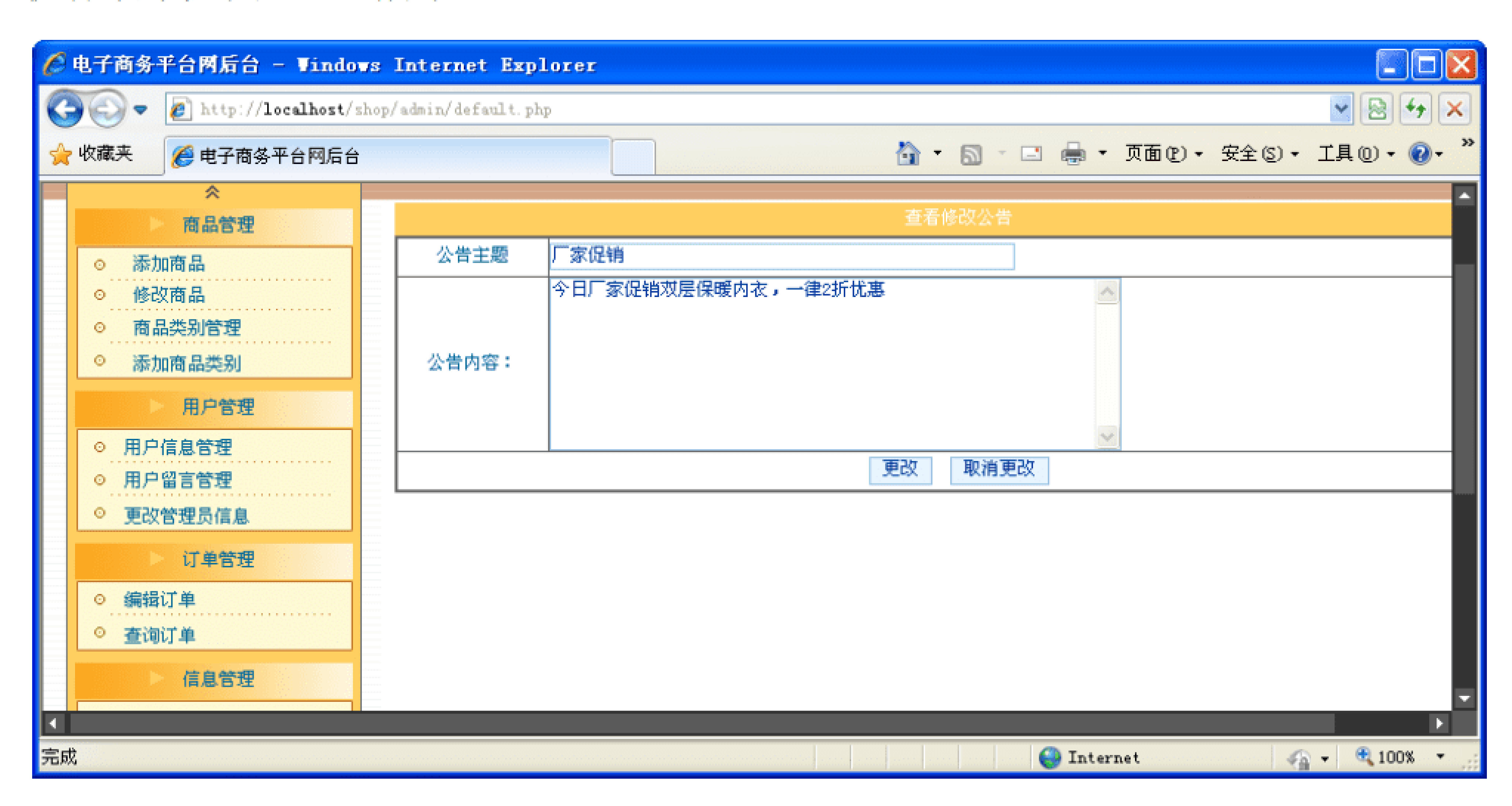


图 15-23　查看编辑公告

2．删除公告

在公告管理页面的公告信息列表中，启用需要删除的公告信息记录的复选框，单击【删除所选项】按钮，即可删除公告信息。实现代码如下所示：

```
<?php
    include("conn/conn.php");       //引入连接数据库文件 conn.php
    while(list($name,$value)=each($_POST))
    //通过 while 循环依次迭代将选中的公告信息记录删除
    {
        mysql_query("delete from tb_gonggao where id='".$value."'",$conn);
    }
    header("location:admingonggao.php");
    //删除成功之后，页面跳转到公告管理页面 admingonggao.php
?>
```

上述代码连接数据库后，通过 while 循环依次迭代将管理员选中的公告信息记录完全删除，删除成功之后页面跳转到公告管理页面 admingonggao.php。

3．添加公告

在后台管理主页中，单击该页面左边的【添加公告】链接，即可进入添加公告页面，该页面定义了一个填写公告信息的表单，管理员可以在该表单中添加公告信息，单击【添加】按钮完成添加公告操作。该页面还对管理员输入公告信息是否合法进行了验证。实现代码如下所示：

```
<script language="javascript">
    function chkinput(form)            //验证表单数据
    {
        if(form.title.value=="")       //验证公告主题
         {
```

```
            alert("请输入公告主题!");
            form.title.select();
            return(false);
        }
        if(form.content.value=="")      //验证公告内容
        {
            alert("请输入公告内容!");
            form.content.select();
            return(false);
        }
        return(true);                   //验证完全通过返回 true
    }
</script>
```

上述代码从表单对象 form 中获取管理员输入的公告信息数据，判断验证管理员输入的数据是否合法，如果合法返回 true，否则返回 false。在此省略了添加公告页面、定义公告信息表单的代码，详细代码可查看随书光盘本章目录中的 shop/admin/addgonggao.php 文件。

当单击【添加】按钮，可以将公告信息数据提交到用于保存公告信息的页面 savenewgonggao.php，实现代码如下所示：

```
<?php
    include("conn/conn.php");       //引入连接数据库文件 conn.php
    $title=$_POST[title];           //获取公告标题
    $content=$_POST[content];       //获取公告内容
    $time=date("Y-m-j");            //设置公告时间
    //发出 insert 语句保存公告信息
    mysql_query("insert into tb_gonggao (title,content,time) values ('$title',
'$content','$time')",$conn);
    echo "<script>alert('公告添加成功!');history.back();</script>";
    //提示公告添加成功
?>
```

上述代码获取表单提交的公告标题、内容，并设置公告发布时间为当前时间，发出 insert 语句将公告信息保存到数据库中，公告添加成功之后，将弹出一个提示添加成功的提示对话框。执行效果如图 15-24 所示。

4．评论管理

管理员单击后台主页左边的【评论管理】链接，可以看到评论管理页面 editpinglun.php。该页面首先从数据库中查询用户对本网站商品的所有评论信息的总条数，计算出每页显示 20 条及需要显示的页数，然后定义一个表格，将从数据库查询到的评论信息数据加载分页显示到表格中。在表格中还定义了一个【查看】链接和一个表单，单击【查看】链接可以查看详细的评论信息。在表单中定义一个【删除选项】按钮，当用户启用评论记录旁的复选框时，单击【删除选项】按钮即可删除评论记录。在此不再给出评论管理页面的代码实现，详细代码实现可查看随书光盘本章目录中的 shop/admin/editpinglun.php。执行效果如图 15-25 所示。

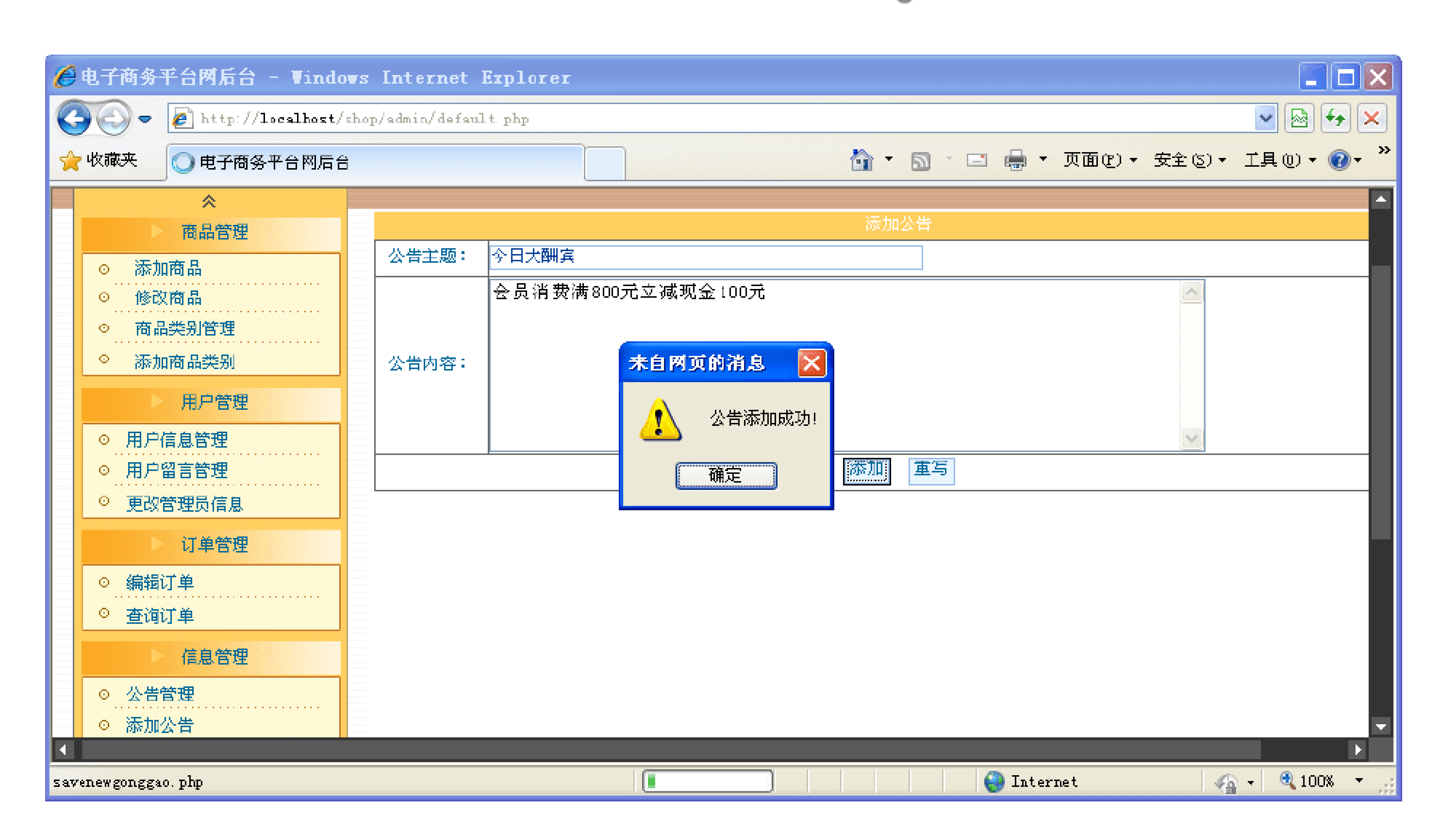


图 15-24　添加公告

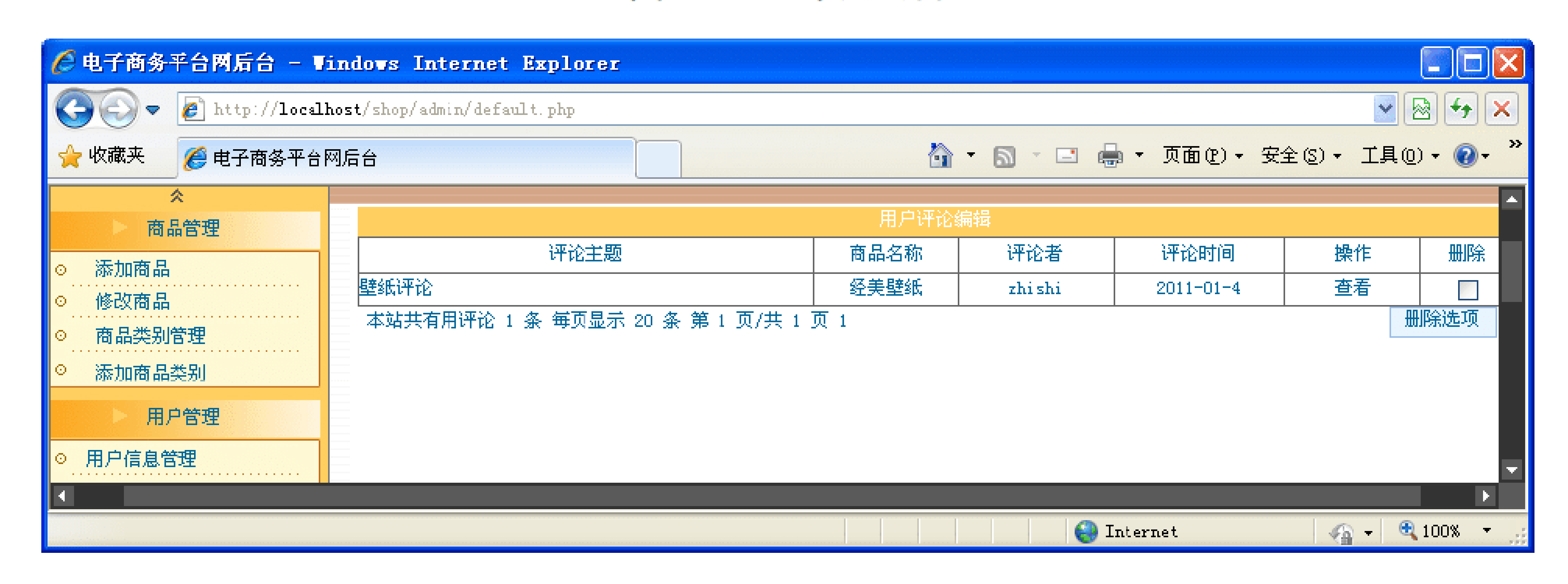


图 15-25　评论管理

当用户单击评论管理页面评论列表中的【查看】链接时，将弹出一个新窗口，在该窗口中加载显示评论的详细内容，实现代码如下所示：

```
<?php
    include("conn/conn.php");        //引入连接数据库文件 conn.php
    include("function.php");         //引入 function.php 文件
    $sql=mysql_query("select * from tb_pingjia where id='".$_GET[id]."'",
        $conn);                      //通过评论 id 查询评论的详细信息
    $info=mysql_fetch_array($sql);   //获取查询结果
?>
```

上述代码通过评论 id 查询评论的详细信息，在查看评论信息页面中，还定义了一个表格，将查询到的评论信息显示在表格中。在此只给出了查询评论代码，详细代码实现可查看随书光盘本章目录中的 shop/admin/lookpinglun.php 文件，执行效果如图 15-26 所示。

删除评论的实现方法与删除用户信息、用户留言大致相同，在此不再详细讲解，详细代码实现可查看项目随书光盘本章目录中的 shop/admin/deletepingjia.php 文件。

至此，一个完整的电子商务平台网已经完成了。此网站是应用 PHP+MySQL 技术实现的，

只是实现了一些常用基本功能，如果你有足够的兴趣，可以再将此系统进行拓展，让这个网站更加完善。

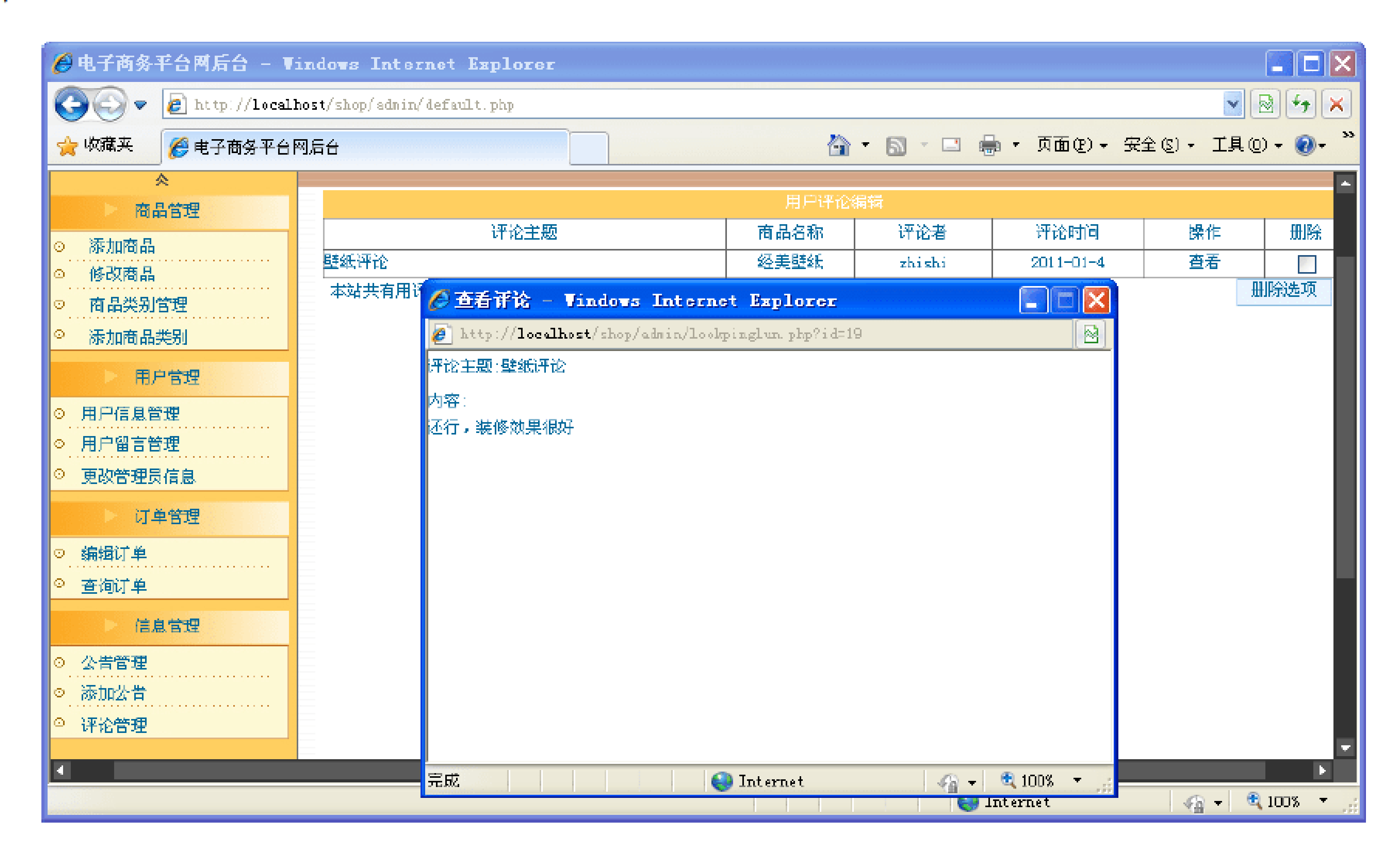


图 15-26　查看评论

习 题 答 案

第 1 章

一、填空题

(1) Hypertext Preprocessor
(2) 2008
(3) 设置全局变量

二、选择题

(1) D
(2) D
(3) C

第 2 章

一、填空题

(1) gettype()，settype()
(2) $
(3) is_null()

二、选择题

(1) B
(2) A
(3) A

第 3 章

一、填空题

(1) const
(2) printf
(3) hello

二、选择题

(1) C
(2) A
(3) D

第 4 章

一、填空题

(1) error_reporting
(2) track_errors(on,off)
(3) ignore_repeated_errors
(4) display_startup_errors
(5) Exception()
(6) log_errors(on,off)
(7) ignore_repeated_source
(8) errorcode
(9) getMessage()
(10) getLine()

二、选择题

(1) B
(2) A
(3) A

第 5 章

一、填空题

(1) strlen()
(2) 字符类
(3) preg_replace_callback()
(4) str_pad()

二、选择题

(1) D
(2) D
(3) D

(4) B
(5) C

第 6 章

一、填空题

(1) readdir.txt
(2) is_readable()
(3) 字符串
(4) fgetcsv()
(5) 0

二、选择题

(1) C
(2) A
(3) C
(4) D

第 7 章

一、填空题

(1) php
(2) $_SERVER['PHP_SELF']
(3) shops[]

二、选择题

(1) B
(2) A
(3) A
(4) B

第 8 章

一、填空题

(1) $_SESSION
(2) session_start();
(3) session_destroy()

二、选择题

(1) C
(2) A
(3) A

第 9 章

一、填空题

(1) mysql_pconnect()
(2) mysql_result($result,0,"username")
(3) mysql_db_name()

二、选择题

(1) A
(2) B
(3) C

第 10 章

一、填空题

(1) safe_mode(boolean)
(2) disable_functions
(3) doc_root
(4) dopen_basedir
(5) user_dir
(6) ServerSignature
(7) 非对称加密
(8) 私有密钥(privatekey)
(9) SHA-1
(10) 128

二、选择题

(1) D
(2) C
(3) D

第 11 章

一、填空题

(1) SMARTY_DIR

(2) $config_dir
(3) capitalize
(4) date_format
(5) FALSE
(6) foreach
(7) foreachelse
(8) insert

二、选择题

(1) D
(2) A
(3) C
(4) B

第 12 章

一、填空题

(1) 元素
(2) replaceChild($newName,$oldName)
(3) 属性

二、选择题

(1) B
(2) C
(3) D

第 13 章

一、填空题

(1) Asynchronous JavaScript And XML
(2) XMLHttpRequest
(3) 服务器接受

二、选择题

(1) C
(2) A
(3) B
(4) C
(5) B